Visual tracking of instruments in minimally invasive surgery

Max Allan

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
of the
University of London.

Department of Medical Physics and Biomedical Engineering
University College London

February 5, 2017

I, Maximilian Allan, confirm that the work presented in this thesis is my own.
Where information has been derived from other sources,
I confirm that this has been indicated in the thesis.

Abstract

Reducing access trauma has been a focal point for modern surgery and tackling the challenges that arise from new operating techniques and instruments is an exciting and open area of research. Lack of awareness and control from indirect manipulation and visualization has created a need to augment the surgeon's understanding and perception of how their instruments interact with the patient's anatomy but current methods of achieving this are inaccurate and difficult to integrate into the surgical workflow. Visual methods have the potential to recover the position and orientation of the instruments directly in the reference frame of the observing camera without the need to introduce additional hardware to the operating room and perform complex calibration steps.

This thesis explores how this problem can be solved with the fusion of coarse region and fine scale point features to enable the recovery of both the rigid and articulated degrees of freedom of laparoscopic and robotic instruments using only images provided by the surgical camera. Extensive experiments on different image features are used to determine suitable representations for reliable and robust pose estimation. Using this information a novel framework is presented which estimates 3D pose with a region matching scheme while using frame-to-frame optical flow to account for challenges due to symmetry in the instrument design. The kinematic structure of articulated robotic instruments is also used to track the movement of the head and claspers. The robustness of this method was evaluated on calibrated ex-vivo images and in-vivo sequences and comparative studies are performed with state-of-the-art kinematic assisted tracking methods.

Acknowledgements

The path that led me to starting this thesis began long ago trying to make hobby robots as an undergraduate, which of course ended in dismal failure. However it took me away from a possible research path in Physics towards Computer Science where I fully began to appreciate exactly how bad a programmer I really was. My MSc project on stereo PTAM with Dr Eddie Edwards was my first exposure to how computer vision and robotics could fit in alongside surgery and I am hugely grateful to him for his guidance and assistance on this project, which I immensely enjoyed. After deciding at the last possible moment that I would like to pursue a PhD he put me in contact with a colleague who was moving to UCL and was working on both computer vision and robotics in surgery, what good fortune! I joined Dr Dan Stoyanov's new surgical robot vision group and over the years I have learned how little I knew, how little I could drink and have hopefully improved slightly in both measures. I feel very lucky to have had Dan as a supervisor, he provided guidance and ideas when my methods inevitably didn't work and had a helpfully no nonsense approach to research, which is surprisingly useful when you're trying hard to make everything as complicated as possible! I was the first PhD student in Dan's group and have seen the group grow to include some excellent researchers who have become good friends. I am grateful to have spent these past 5 years working alongside Martin, Geoff, Xiaofei, Rene, Ping-Lin, George, Evans, Francisco, Francois, Krittin and Mirek who have made the time all the more enjoyable and I hope to remain good friends with all of them in the years to come.

The work in this thesis could not have been completed without collaboration with other researchers at CMIC. I am grateful to Matt Clarkson and Steve Thompson for their help with experiments on my first papers and allowing me to avoid having to learn how to use an Optotrak. I also am grateful to Professors Sebastien Ourselin and Dave Hawkes for being my secondary supervisors, reading my paper drafts and providing comments and guidance. The data used in this thesis was dependent on the surgeons at UCLH, Professor John Kelly, who was also one of my supervisors, and Dr Ashwin Sridhar, they both always accommodated my trips to observe robotic procedures and collect data as well as providing ideas and perspective on what I was trying to achieve. Simon Di-Maio at Intuitive Surgical and the whole team working on the DVRK at JHU have made my research much more painless by providing CAD models, instruments, software and hardware to collect data. I am also grateful to Austin Reiter at JHU, Zachary Pezzementi at CMU and Menglong Ye and Lin Zhang at ICL for providing comparative data which enabled me to validate my methods. I also cannot forget my time at Siemens Corporate Research, where I worked with Dr Peter Mountney for 6 months after my MRes year. This was my first exposure of research in the corporate world and it was hugely enjoyable. I am very thankful to have been able to experience working in a new environment and my time there improved my programming ability tenfold. I arrived having never used a smart pointer and left having written a shader and experienced the perils of trying to use multithreading with OpenGL. Finally, I would like to acknowledge my thesis examiners, Professor Lourdes Agapito and Professor Raphael Sznitman who provided an enjoyable and insightful examination and gave me many useful suggestions for improving my future work.

Without a life outside of UCL, I don't know if I would have made it to the end of this thesis. My first thanks go to my fiancée Sne, who has tolerated less than enjoyable weekends and weeknights watching me fix bugs at my laptop rather than spending our time together doing something normal and fun. Her constant support has helped me to feel that I wasn't wasting my life and pushed me through at times when it might have been easier to give up. I cannot express how much I'm looking forward to our wedding in India next year and to a long and happy life together. I am also thankful to my friends Orestis, Sofia, Tom, Euan and Alistair, although we almost never find the time to see each other any more I feel that might be about to change. This is with the exception of Euan who lives about as far away as possible while still remaining on this planet, we'll have to stick to the chat clients on online games. I am also grateful to my parents, Patsy and John and my sister Lizzie who I also don't see nearly enough but have always made me feel proud about what I do.

After finishing this thesis I am leaving UCL for industry and I imagine life will become a little more stressful and perhaps have a few more boring meetings. I know that during these meetings I will look back fondly at these past few years of working on fascinating problems in whatever way I wanted and think it all went by a bit too fast.

Contents

Figures

Tables

Acronyms

MIS	Minimally Invasive Surgery
OR	Operating Room
NOTES	Natural Orifice Transluminal Endoscopic Surgery
CAS	Computer Assisted Surgery
CAI	Computer Assisted Intervention
RMIS	Robot Assisted Minimally Invasive Surgery
HoG	Histogram of Oriented Gradients
SIFT	Scale Invariant Feature Transform
SURF	Speeded-Up Robust Features
IID	Independently and Identically Distributed
ML	Maximum Likelihood
LND	Large Needle Driver
RF	Random Forest
ND	N-Dimensional (where N is an integer)
SSD	Sum of Squared Difference
SVM	Support Vector Machine
CNN	Convolution Neural Network
PnP	Perspective N Point
DoG	Difference of Gaussians
LoG	Laplacian of Gaussians
GLSL	OpenGL Shader Language
DVRK	da Vinci Research Kit
MTM	Master Tele Manipulator
PSM	Patient Side Manipulator
ROS	Robot Operating System

RCM	Remote Center of Motion
WP	Wrist Pitch
WY	Wrist Yaw
SUJ	Set Up Joints

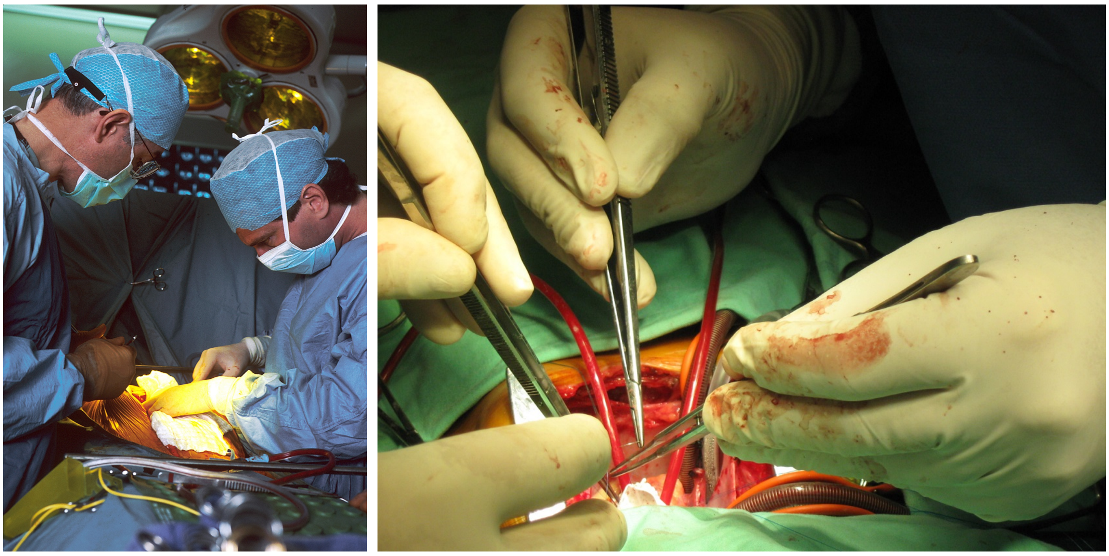
Chapter 1

Introduction

1.1 Open Surgery

For decades, open surgery has been the preferred method for a surgeon to access patient anatomy. The surgeon creates an incision large enough to allow them to directly view and manipulate the tissue using their hands and instruments such as scissors, forceps and scalpels. Such direct access gives the surgeon control of the tissue, tactile cues enable the surgeon to understand and locate vessels, ducts, tumors and other abnormalities and direct visual observation provides information about the shape, size and location of anatomical structures.

The type, size and number of incisions required in open surgery varies between different procedures. In abdominal procedures such as retropubic prostatectomy, an incision of around 10 cm (see Figure 1.2a) can be made in soft tissue from the umbilicus to the pubic area and access to the prostate is provided with separation of the abdominal muscle. Much more invasive techniques are required during procedures where the anatomy is obstructed by bone or critical vasculature. For example in open oral and maxillofacial surgery, where the surgeon is attempting to access intraoral and oropharyngeal lesions, the normal access route is created through a mandibulotomy (see Figure 1.2b) whereby the surgeon divides the lower lip, mandible and tongue [12]. In cardio-thoracic surgery, the most common access route is a median sternotomy where a 10 - 15 cm incision is made in the chest wall and the sternum is severed and opened with a surgical saw.



(a) (b)

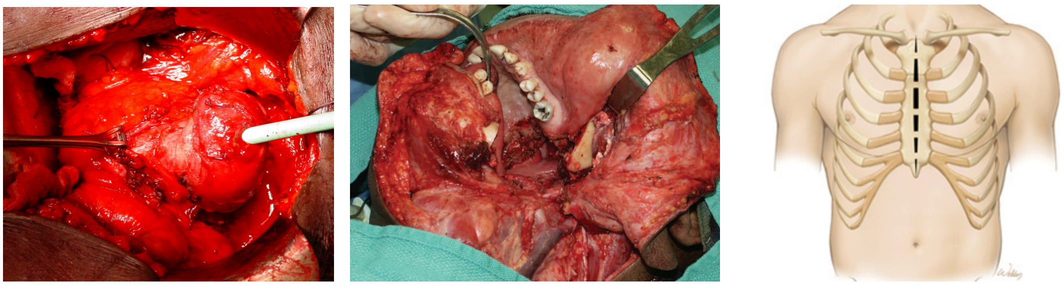
Figure 1.1: (a) The operating table for an open procedure. (b) The instruments used during an open heart surgery.

The invasiveness of open surgery is a hugely significant problem with numerous medical, cosmetic and economic consequences. One of the major medical drawbacks of open surgery is the quantity of blood lost by the patient. The length of the procedure as well as the tissue trauma required means that procedures such as a retropubic prostatetectomy have a mean blood loss of between 600 and 1500 ml [13] which in more extreme cases can cause the patient to go into shock due to reduced blood pressure and may require post-operative blood transfusions. Additionally, during many open procedures, internal organs are regularly cooled, dried, handled and retracted [14] which can cause complications such peritoneal adhesions, which can form on the abdomen wall during abdominal procedures. In addition to intra-operative complications, the consequences of open surgery also pose risks for the patient post-operatively. The large incisions required in open procedures carry the risk of wound infection, which occurs in between 1 to 10 % of all procedures [15] and is the second most common cause of hospital acquired infection. Other wound related complications such as dehiscence, cellulitis and incisional hernia [16] are all impacted directly by the trauma level of the procedure. In the potentially lengthy recovery period the patient is left immobile in a hospital bed where muscle inactivity can cause pulmonary and cardiac complications and deep vein thrombosis [17]. Additionally if the procedure requires a large quantity of analgesia, which is often the case for lengthy and complex procedures, the after effects of the drugs in the patient's body can cause numerous undesirable side-effects such as respiratory depression and hypotension [18]. Cosmetic complications related to highly traumatic open procedures are also a significant disadvantage of open surgery. Large scarring and, in cases where bones are cut, delayed or incomplete unification of the fragments [12] can all have impacts on a patient's post operative appearance.

There are also economic challenges, from the increased cost of the patient's recovery time in the hospital to their extensive leave from regular employment. The financial impact of the post-operative recovery period is far reaching and felt across different scales from the individual patient to all the way up to significant macroeconomic impact. Typical inpatient recovery for an open radical prostatectomy can range from 4 to 7 days with more invasive procedures such as a mandibulotomy requiring a period that can stretch into several weeks [19]. Returning to normal activity levels can take up to 3 months for a median sternotomy [20]. Patients who spend large amounts of time out of the workforce are more likely to drop out of employment permanently, even once their medical conditions have abated. This has implications for their financial status as well as their long-term mental health [21]. Viewed from the perspective of a healthcare provider, this situation is less than ideal as the long-term occupation of a hospital bed adds to operation cost by hundreds of pounds per day and post-operative complications involve additional costs [22] all of which limit the pool of money available for other procedures. Finally, on a national scale large numbers of long-term hospital stays and lengthy recovery periods are an economic burden as unemployment or sick-leave benefits must be paid and additionally businesses must cope for longer with missing or substitute staff.

1.2 Minimally Invasive Surgery

Pain and morbidities such as discomfort and disability that result from open surgery are most commonly caused by the trauma in gaining access to the surgical site rather than from the procedure itself, for example in a cholesystectomy the need for post-operative hospitalization is entirely caused by the trauma in the abdominal wall which is cut to gain access to the gallbladder [23]. Reducing the access trauma of open procedures through Minimally Invasive Surgery (MIS) has become one of the most significant developments of modern medicine [24]. It has influenced techniques in almost all aspects of surgery leading to the complete replacement of many open techniques and the modification of many conventional procedures with respect to improving patient experience [25]. The main idea behind MIS is to forego



(a) **(b)** **(c)**

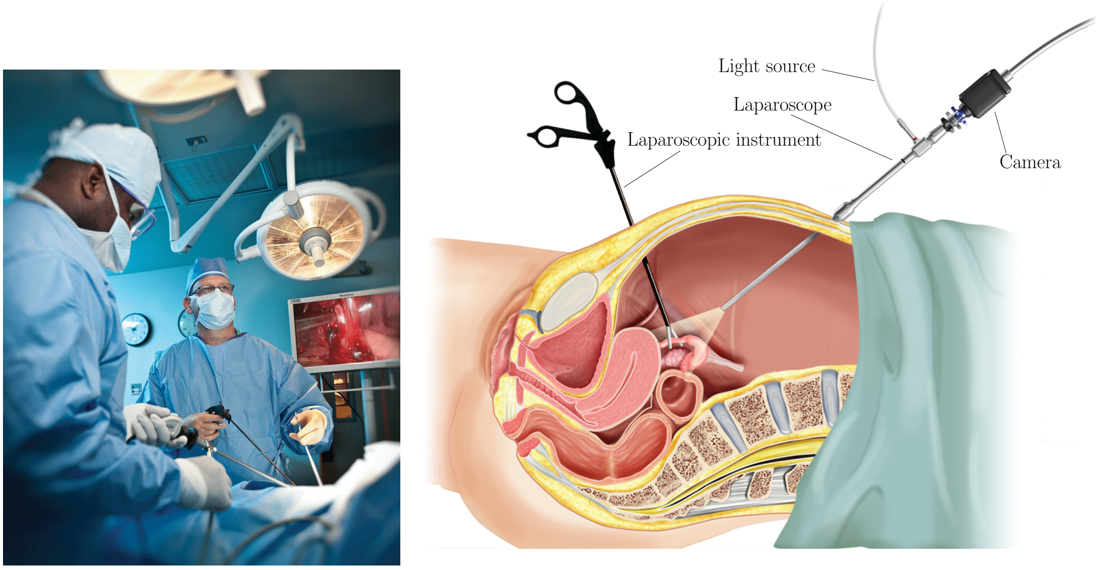
Figure 1.2: (a) An open incision during a retropubic prostatectomy. This is a comparatively small incision compared with other open procedures. (b) A lip-splitting incision in a mandibulotomy [1]. The patient's entire mandible, lip and tongue are severed to provide access to the oral cavity and oropharynx. (c) A meridian sternotomy which is used in many thoracic operations to provide access to the thoracic cavity [2].

attempting to directly visualize and interact with the anatomy and instead try to minimize the access trauma either by creating small ports in the patient's body or by using natural orifices. If required, the access ports are created to be wide enough to allow elongated instruments and a laparoscope to pass through enabling the surgeon to manipulate the tissue while his or her hands remain outside of the body. The entire procedure is observed on a video feed from the laparoscope on a display in the theatre either in 2D or, more recently, in 3D (see Figure 1.3).

(a) **(b)**

Figure 1.3: (a) Surgeons viewing a typical 2D display while they operate. (b) The same situation with 3D displays which create the stereoscopic effect using polarized glasses.

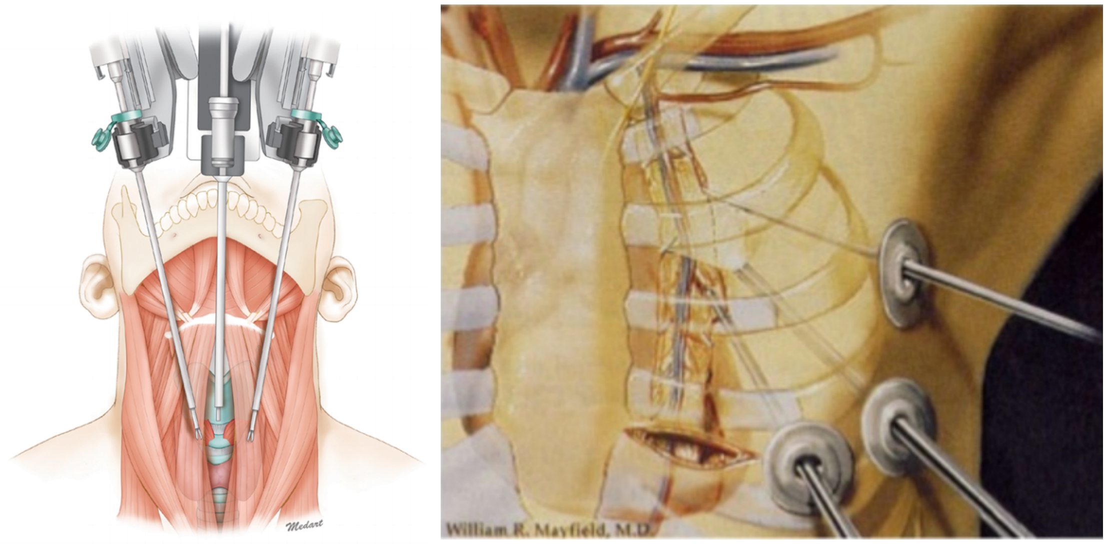
The technological developments which enabled the growth of MIS began around 200 years ago with the early endoscopes of Philip Bozzini. He demonstrated these as cystoscopes and vaginoscopes but it look until the mid 1800s before advancements in light sources and lens systems of Antonie Jean Desormeaux and Adolf Kussmaul received more widespread acceptance in the medical community. In the early 20th century, Hans Christian Jacobaeus introduced the idea of laparoscopy and further technological developments throughout the 20th century showed that the techniques could be used in different types of surgery. The rod-lens system of Harold Hopkins [26] was a huge leap forwards as it increased light transmission by around 80 % and fibre-optic technology developments in the mid 20th century enabled much clearer visualization and increased interest for further development from surgeons themselves. This led to changes and improvements to surgical practice, planning and instrument design and miniaturization [27] and by the 1960s arthroscopies were the preferred method to diagnose and treat maladies of the knee [28]. In the 1990s, the development of CCDs and high resolution video cameras which could be attached to an endoscope allowed the entire OR staff to simultaneously view a clear, magnified image of the procedure and in 1987, Phillipe Mouret demonstrated the first laparoscopic cholecystectomy from which the whole field took off [23]. Laparopscopic cholecystectomy revolutionized abdominal treatments almost completely replacing conventional surgery and led to virtually all gastrointestinal tract surgery being minimally invasive. Shortly after Mouret's demonstration, laparoscopy became used in colectomy, splenectomy, nephrectomy, adrenalectomy, appendectomy, small bowel resections and explorations amongst others. Improvements in skills and technology led to minimally invasive versions of previously highly challenging procedures such as gastric bypass, hepatoportoenterostomy, total abdominal colectomy and esophageal atresia repair [27]. By the turn of the century, a majority of urology and thoracic surgeries in the United States were being performed with minimally invasive techniques.



(a) **(b)**

Figure 1.4: (a) The operating room for a laparoscopic procedure. The surgeon watches the procedure on a raised display which breaks the hand-eye coordination normally present during open surgery. (b) The set up for a laparoscopic surgery. The patient's abdomen is insufflated with CO_2 and instruments and a camera are inserted through *trocar* insertion points.

The advantages of this type of procedure for the patient are enormous. The small or non-existent access incisions significantly decrease tissue and bone trauma compared with equivalent open procedures. For instance, instead of the highly traumatic mandibulotomy (see Figure 1.2b), a minimally invasive alternative involves a trans-oral procedure where the surgeon acquires access via the patient's open mouth and throat completely avoiding the need to sever the lips, tongue and mandible. In this case, no exterior incision is required for access and this is known as natural orifice translumenal endoscopic surgery (NOTES) where the technique has also been applied for abdominal procedures making use of the urethra, anus and vagina for access [29]. In minimally invasive cardiac surgery, certain procedures can be performed using a mini thoracotomy where the surgeon is not required to break the sternum and instead makes several small incisions in the right hand side of the patient's chest passing the instruments through the muscles between the ribs (see Figure 1.5). This minimizes patient post-operative pain and the reduced incision size and trauma results in a reduction in the likelihood of post-operative infection [15]. In addition to this, blood loss during the operation is decreased by up to half the quantity lost in open procedures [13, 30]. This also has the effect of increasing the range of patients and procedure types which are viable. Open surgery is often too traumatic for elderly or infant patients, and for mild conditions such as obstructive sleep apnea, an open mandibulotomy is far too traumatic to justify. However, when addressed minimally invasively the procedure becomes an attractive option. The longer term medical outcomes such as mortality and resection margins, which affect tumor recurrence, have been shown in numerous studies for various MIS procedures to have comparable performance to open counterparts [31, 32, 33, 30]. Economic limitations of open surgery center primarily on the inpatient and outpatient recovery time. The huge reduction in trauma from MIS procedures dramatically reduces recovery time with mean hospitalization for colonic resection reduced from 7-9 days for open procedures to 2-3 days for minimally invasive alternatives. This leads to a faster return to work with a 37.5 day reduction for coronary revascularization, a 9 day reduction for prostatectomy and a 16.6 day reduction for peripheral revascularization [34]. Additionally, health plan spending has been shown to be significantly lower in many MIS procedures with between $1500 in yearly savings for uterine fibroid resections and $30000 for coronary revascularization [34]. The cosmetic improvements of a surgical technique are harder to measure as they involve subjective patient assessment. However, a study on the appearance of a 1 month old MIS thyroidectomy wound and a 1 month old conventional thyroidectomy wound found that the minimally invasive procedure produced statistically significant preferential results [35].



(a)

(b)

Figure 1.5: (a) A minimally invasive trans-oral procedure which enables the surgeon to access the mouth and throat without traumatic injury to the patient as seen in Figure 1.2b. (b) The incisions made when accessing the heart in a minimally invasive cardiac surgery. This does not necessitate breaking the sternum as in a sternotomy [3].

Despite the advantages of MIS, there are many complications introduced by the impedance of direct visual and tactile access to the patient's anatomy. During a minimally invasive procedure the surgeon is no longer able to see the entire anatomy while they work which reduces navigation ability and awareness as the endoscopic camera has a highly restricted field of view and can easily become occluded by blood, smoke or instruments. In addition to this, the surgeon watches the operation's progress on a raised display unit which is often located in a position that disrupts the hand-eye coordination between the motion of the surgeon's hands and the observed motion of the instruments. Alongside the visualization challenges, the restriction of direct access to the patient's anatomy has several significant limitations for the surgical team [36]. Impeding physical contact prevents the surgeon from using the tactile cues provided by handing tissue in open surgery and additionally the instrument design virtually eliminates useful haptic feedback which the surgeon normally uses to understand tool-tissue interactions which can lead to dangerous forces being applied to critical structures. Despite advances in control methods for laparoscopic instruments, they still impact the surgeon's dexterity when interacting with tissue. The length of the instrument as well as the inversion of translational motion caused by its pivot around the trocar insertion point leads to difficulty in performing precise movements or controlling the anatomy (see Figure 1.4b). Additionally interacting with the anatomy using a single joint grasper limits the surgeon's ability to perform complex or challenging techniques during the operation which in turn limits the complication of procedures which MIS can address. Despite the standardization of training for new laparoscopic surgeons in specialized training centers (see Figure 1.6), a final limitation around the ongoing transition from open surgery to MIS revolves around how pre-existing expertise in surgical practice can be adapted for minimally invasive procedures. Until very recently, the majority of senior surgeons have years of experience operating using open methodologies and as the techniques required for general surgery do not translate directly to MIS they must retrain extensively if they are to bring their expertise to the field. Surgeons may only be required to learn basic laparoscopic control skills for simpler procedures such as laparoscopic cholsystectomy but to perform the vast majority of more complex procedures they also have to learn advanced skills such as coordinated two-handed dissection, retraction, suturing and the use of new instrumentation [37]. A consequence of the lack of consistent and comprehensive training programs for assisting surgeons in making the transition from open surgery to MIS has led to an increase in complications [38] and how to address this with objective and comprehensive training programs is an open area of research [39].

Figure 1.6: The Chitra Sethia Centre for Robotics and MAS in London, UK where surgical training programs are provided to assist surgeons in obtaining laparoscopic and robotic surgical skills.

1.2.1 Robotic Minimally Invasive Surgery (RMIS)

One of the most promising solutions to the problems that have so far limited MIS has been the introduction of robotics to the operating theatre. Medical robotics has mainly focussed on improving MIS by providing the surgical team with greater control over the imaging sensors and instruments used during the procedure. Hand held minimally invasive instruments are often difficult to control precisely and robotic systems have been used to provide more intuitive or dexterous manipulation methods [40]. The first application seen in the OR was an industrial PUMA 200 that was re-purposed for clincal use in the 1980s to guide a needle tip in a stereotactic brain surgery [41] due to this procedure's requirement of a very precise straight-line trajectory into the patient's brain to avoid critical brain regions and vessels. As the benefits of the advanced control systems provided by robotics became more widely accepted in the medical community, more procedures were assisted by robotic technology and in the 1990s the PUMA 560 was used in a tranurethreal prostate resection [42] and led to later developments such as Probot ® [43] and ROBODOC ®, which was the first FDA approved surgical robot. These early systems were typically focussed on a single surgical procedure but interest in telepresence surgery from NASA scien-

(a)

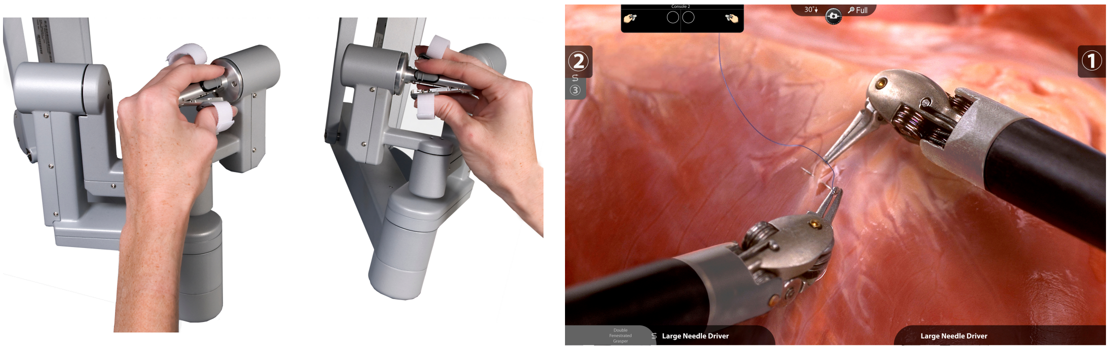
(b)

Figure 1.7: (a) A ROBODOC system which was an early orthopaedic robotic system used in arthroplasty. (b) A da Vinci Si HD Surgical System showing the master console and the robot.

tists and later the US military led to the development of more general systems in the form of commercial ventures such as Zeus from Computer Motion, Inc. and the SRI Green Telepresence Surgery System or da Vinci ® from Integrated Surgical Systems (now Intuitive Surgical) [44]. All of these systems were master-slave designs with multiple arms controlled from a central console where video images are displayed. These two companies merged in 2003 and the Zeus design was phased out in favour of the da Vinci which is the most popular system in production today. Its usage has grown from less than 1000 in 2002 to 650000 globally in 2015 with 3597 systems in clinical use around with world [45]. The reasons for the huge uptake of systems such as da Vinci have centered on their improvement of the surgeon's visualization with 3D vision as well as control and dexterity of the instruments and camera while operating. The control system provides the surgeon with dexterous master tele-manipulators (MTMs) (see Figure 1.8a) which enable precision control over the instruments as well as removing the chopstick effect and tremor that make controlling laparoscopic instruments so challenging. These MTMs are connected to articulated instruments on robotic patient side manipulators (PSMs) (see Figure 1.8b) which due to their flexible wrist design enable the surgeon to precisely control the anatomy in a manner that closely mimics direct hand control. The console design of telerobotic systems such as da Vinci also provide the surgeon with a much more comfortable seated environment in which to work, which is significant for lengthy surgeries, and additionally allows the visualization system to be placed naturally in front of the surgeon's eyes creating the effect of virtually placing his or her hands in the place of the instruments. This creates a much stronger visual coupling between the motion of the surgeon's hands and the corresponding motion of the instruments when operating.

1.3 Computer Assisted Interventions

Despite the improvements to control and visualization provided by robotics platforms, there are still many challenges that must be solved before the full potential of MIS can be realised. Concurrently to robotics, the integration of imaging and tracking technology as part of computer assisted interventions (CAI) has greatly assisted surgeons in solving navigation and guidance issues that have arisen through operating minimally invasively [46]. It has been common in operating theatres since the early 1990s and normally first involves the construction of pre-operative patient models from 3D imaging data around which surgical planning can be performed. This enables surgeons to estimate optimal trajectories through the patient's anatomy that minimize trauma to critical structures and to understand and annotate the various pathologies such as tumors that have been detected during biopsies or scans [47]. During



(a)

(b)

Figure 1.8: (a) The master manipulators used by a da Vinci operator to control the instruments. (b) The instruments interacting with a phantom anatomy.

the operation, the annotated pre-operative models are aligned with intraoperative imaging coordinate systems so that the surgeon can make use of the predetermined trajectories and annotations in the plan. MIS also provides an excellent delivery platform for non-white light modalities such as ultrasound, MRI and CT as the data can in principal be fused directly with the video feed that the surgeon uses in the operation. However, finding the correct rigid body transforms between the camera coordinate system and the coordinate system of the imaging modality as well as correcting for non-rigid deformations are extremely complex problems compounded by the lack of cross modality landmarks and visual occlusion from smoke, blood and instruments. Real-time alignment between the coordinate systems would allow pre-operative planning data such as tumor annotations and vasculature to be displayed clearly to the surgeon. Additionally there are complexities with how to render the models onto the visual feed without obscuring instruments and other devices placed into the patient's body.

1.3.1 Visual Tracking of Instruments

Many technical challenges remain before there can be a complete integration of computer assistance in the operating theatre and its benefits can be fully realised. One component to finding a complete solution to this highly complex set of problems relies on having a good understanding of the physical relationship between the different instruments the surgeon has introduced to the patient's body and the imaging sensors and anatomy. This can be used to properly understand the surgeon's interaction with tissue surfaces, improving patient safety and potentially enabling haptic feedback to be synthesized. In addition to this, when working on robotic platforms, soft motion constraints known as virtual fixtures can be applied to enforce a safe distance to pre-operatively defined vulnerable structures such as arteries that the surgeon may wish to avoid [48] and automatic motion guided by the imaging sensor, known as visual servoing, can be used to ease the cognitive load on the surgeon when performing routine tasks [49]. Beyond the typical improvements to the surgical workflow, understanding the instrument position and orientation (known as pose) has important consequences for the assessment of surgical skills [50]. It can enable the precise measurement of the surgeon's motion patterns during operations and surgical training allowing immediate feedback on areas of strength and weakness in methodology. It can also quantitatively assess the impact of new technologies as they are introduced to the operating room (OR) by objectively measuring the influence they have on the surgeon's motion patterns and performance. Currently this is assessed manually which introduces biases in the feedback process [51].

External optical and electromagnetic tracking markers [52, 53] are often used as part of image guidance to track the instruments in real time (see Figure 1.9) but these introduce line-of-site problems and have limited accuracy within complex calibration procedures. Alternatively, in robotic surgery motor

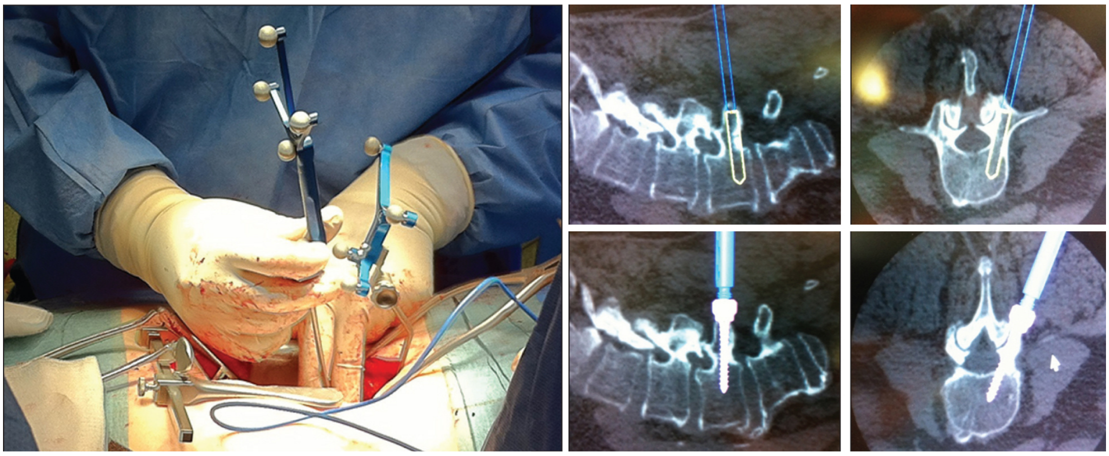


Figure 1.9: An overview of a CAS system for orthopaedic surgery which, through optical tracking of the instruments can provide an overlay on an xray image showing the position of the instruments relative to the anatomy. Image modified from [4].

encoders can provide positional data when the kinematic structure of the robot is known but as with external trackers, calibration is a significant issue and clinical translation is limited as robotics systems outside of the research environment do not typically have accessible joint data. An alternative solution to estimating the pose of instruments is to directly use the images captured by the observing camera. This eliminates the need to attach tracking markers to the instruments and directly reports measurements in the reference frame of the camera, which is the most commonly desired reference frame, without complex hand-eye calibrations. However, solving this problem (known as visual tracking) is highly challenging due to the restricted field of view, occlusions, fast motion of the instruments and a highly deformable and dynamic environment.

1.4 Thesis Overview

The purpose of this thesis is to develop vision based 3D tool tracking without introducing additional hardware into the operating theatre. This means that our primary focus is on solving the problem using only the images captured from the monocular or stereo laparoscopes used in minimally invasive surgery, however as the kinematic data from the robot can be useful and does not require new hardware we will take a relaxed assumption that it can be used to support algorithms we develop for increased performance but the algorithms themselves should be able to track reliably in isolation.

For the methods investigated to be practically useful for the RMIS and CAS the error in this estimation should ideally be submillimeter, and maintained with minimal error over extended sequences during periods of occlusion and noise. As this task is highly complex and far beyond the scope of a single thesis, we instead aim to reduce the error in our pose estimations as much as possible. The desired outcome being that that work in this thesis can be built upon to come closer to the end goal of submillimeter, long-term 3D tracking.

In **Chapter 2** the current state of the art in instrument detection and tracking is discussed. This chapter introduces the various methodologies that have been used to solve the problem and how ideas have been introduced and discarded by the community over time. This will also cover the current limitations in the state of the art which will guide the remainder of the content in this thesis. **Chapter 3** will address the challenge of determining which features can be reliably detected in surgical images (see Figure 1.10b). The computer vision and medical imaging literature has explored a wealth of different features all of which come with different strengths and limitations. Which subset of these features is most appropriate for instrument pose estimation and tracking is an open question and extensive analysis

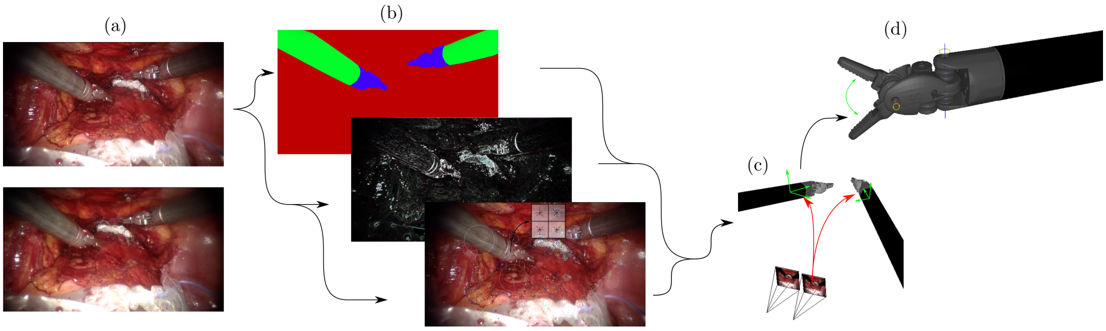


Figure 1.10: The overview of the thesis. (a) Images are captured with a stereo endoscope. (b) Feature spaces are generated on which detection is performed. (c) Rigid pose is estimated for the instruments. (d) The articulated degrees of freedom of the instrument are tracked.

of how these features perform in MIS is needed to make a well reasoned selection. Using these detected features the second challenge which makes up the content of **Chapter 4** is to develop algorithms which can reliably estimate the rigid instrument pose in 3D. This corresponds to the rotation and translation from the camera coordinate system to the instrument coordinate system, ignoring all deformations of the instrument shape due to articulation. Ignoring articulation makes the problem simpler and allows different techniques to be explored more efficiently. **Chapter 5** addresses particular challenges that arose during the analysis in Chapter 4 and **Chapter 6** describes the process of extending the framework developed in Chapters 4 and 5 to track the articulated joints of da Vinci robotic instruments.

1.5 Thesis Contributions

In this thesis the main contributions are the technical methods, which consist of thorough and rigorous evaluation of the features that can be used to drive instrument detection and the formulation of a 3D pose estimation framework that is driven by these features. This framework is easily adaptable to any surgical instrument, assuming the 3D shape is known beforehand and extends to more complex articulations.

The work in this thesis has contributed to the following publications:

1. M. Allan, S. Ourselin, S. Thompson, D. J Hawkes, J. Kelly, D. Stoyanov. Towards detection and localization of instruments in minimally invasive surgery. *IEEE Transactions on Biomedical Engineering*, 60(4), pp. 1050-1058 (2013)

2. M. Allan, S. Thompson, M. J. Clarkson, S. Ourselin, D. J. Hawkes, J. Kelly, D. Stoyanov. 2d-3d pose tracking of rigid instruments in minimally invasive surgery. *Information Processing in Computer-Assisted Interventions*, pp. 1-10 (2014)

3. M. Allan, P. L. Chang, S. Ourselin, D. J. Hawkes, A. Sridhar, J. Kelly, D. Stoyanov. Image based surgical instrument pose estimation with multi-class labelling and optical flow. *Medical Image Computing and Computer Assisted Interventions*, pp. 331-338 (2015)

4. X. Du, M. Allan, A. Dore, S. Ourselin, D. Hawkes, J. Kelly, D. Stoyanov. Combined 2D and 3D tracking of surgical instruments for minimally invasive and robotic-assisted surgery. *International Journal of Computer Assisted Radiology and Surgery*, 11(6), pp. 1109-1119 (2016)

5. K. Pachtrachai, M. Allan, Vijay Pawar, Stephen Hailes, D. Stoyanov. Hand-eye calibration for robotic assisted minimally invasive surgery without a calibration object. *International Conference on Intelligent Robots and Systems*, pp. 2485-2491 (2016)

6. D. Bouget, M. Allan, D. Stoyanov, P. Jannin. Vision-Based and Marker-Less Surgical Tool Detection and Tracking: a Review of the Literature. *Medical Image Analysis*, 35, pp. 633-654 (2016)

Chapter 2

Instrument Detection and Tracking in Minimally Invasive Surgery

2.1 Introduction

Like many problems in computer vision, the visual detection and tracking of instruments can be posed as a parameter estimation problem which can be solved with machine learning or statistical modeling techniques. This can be performed in a supervised learning framework whereby a human or gold standard provides the desired output and the parameters are estimated which minimize some distance measure between the predicted output and the desired output. Alternatively the problem is posed within a model fitting framework where a generative model is used to predict the observed image data and the parameters which cause the generative model to more accurately describe the observed image data are estimated online.

There are two main domains for this, in 2D where the instrument is represented by a set of parameters in the image plane, such as a bounding box where the estimated parameters are center, scale and rotation or alternatively in 3D where the parameters are the 3 Euler rotation angles, or an equivalent representation, and translation. In both cases articulated instruments may be represented by additional parameters. There are three main stages to a general visual tracking method in computer vision: the identification of image features which can be used to estimate the pose of the target object from a single image; identifying the correct pose for a given frame given a set of image features; and maintaining a consistent estimate of the pose over an extended sequence of images, accounting for possible occlusions and failure cases. To gain a broad theoretical understanding of these problems and their solutions, the general computer vision literature is a natural starting point. However, our target application of MIS/Robotic surgery introduces several quite specific challenges that are not normally considered within the computer vision community, and this requires non-trivial modification to be made to these techniques to enable them to work within our target environment. The work in this chapter comprises a section of the publication [54].

2.2 Features for Instrument Detection

The basic aim of detection methods is to find the parameters $\boldsymbol{\theta}$ that describes the object's position and orientation or pose in an image or a sequence of images. This can be achieved as a distribution estimate over the parameters or alternatively point estimates of the parameters can be found. The computational cost of estimating these parameters by processing an entire image in one function would be enormous so various simplifications are often made to process the image more efficiently. To solve this problem, two main steps are taken. The first is to break the image down into either single pixels or groups of pixels and assume that these subsections of the image are independent from one another. This enables them to

	Features				**Pose Estimation**				
Method	**Color**	**Texture**	**Multiple Images**	**Labels**	**Dimensionality**	**Generative**	**Discriminative**	**Algorithmic**	**Temporal**
[55]	RGB			N. Bayes	2D	Region			Initialization
[56]	HS			Threshold	2D			×	
[57]	HS			N. Bayes	2D	Region			Particle
[58]	Sat			Threshold	2D			×	
[59]	RGB				3D	Template			Kalman
[60]	RGB			N. Bayes	2D	Region			Particle
[61]	HS	Sobel						×	
[50]	HS			N. Bayes	2D			×	Particle
[62]	HS				3D			×	
[63]		Gradient			3D	Edges			
[64]	HS	Sobel			2D	Edges			
[65]	RGB	Sobel	Motion		3D	Edges			
[53]	HS		Depth	N. Bayes	2D			×	
[66]	Norm Red			N. Bayes	2D			×	
[9]	ConeHSV	Haralick		N. Bayes	3D	Region			
[67]	Gray				2D	Template			Initialization
[68]	RGB				2D			Active Testing	
[69]		Sobel			3D	Edge			Particle
[70]	HSV	Gradient, Hessian			3D	Point	Kalman		
[71]		SIFT, Haralick		N. Bayes	3D	Line-Mod			
[72]	Gray	Gradient			2D	Template	Adaboost		Initialization
[73]	RGB				2D	Template			Initialization
[74]	RGB	HOG	Motion		2D	Point, Region	L-SVM		Initialization
[75]	Gray		Motion, Disparity	N. Bayes	2D			×	Particle
[76]	RGB				2D	Median Flow	Cascaded Ferns		Initialization
[77]		Gradient			2D		Gradient Boosting		
[78]		Gradient		Threshold	3D			×	Kalman
[79]	RGB,HSV, CIElab	Gradient		RF	3D			×	
[80]	RGB, CIELab			Threshold	2D			×	Initialization
[81]	Gray	HOG			2D		RF		Initialization
[82]	CIELab	HOG, Texton			2D		SVM		
[83]	HSV, CIELab, Opponent	LBP		RF	2D			×	

Table 2.1: An overview of the methods covered in the review. The features used are shown in the first 4 columns, then the type of pose estimation in columns 4-8 where we additionally specify if the method estimated 3D instrument pose or 2D instrument pose. Algorithmic methods are not specified by name as they do not fall under an umbrella term, with the exception of active testing. Finally in column 9 the tracking technique used is indicated.

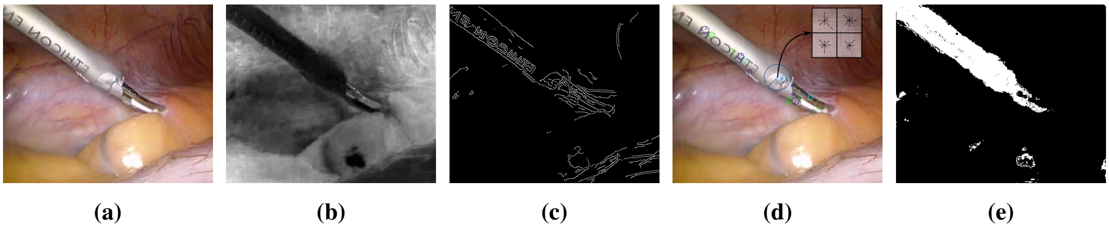

(a) (b) (c) (d) (e)

Figure 2.1: Examples of image features. (a) An image from a typical minimally invasive procedure captured through a laparoscope. (b) The frame transformed into the saturation color space, which is often effective at highlighting metallic objects. (c) Edge features. (d) Extracted texture features. (e) A semantic labelling map.

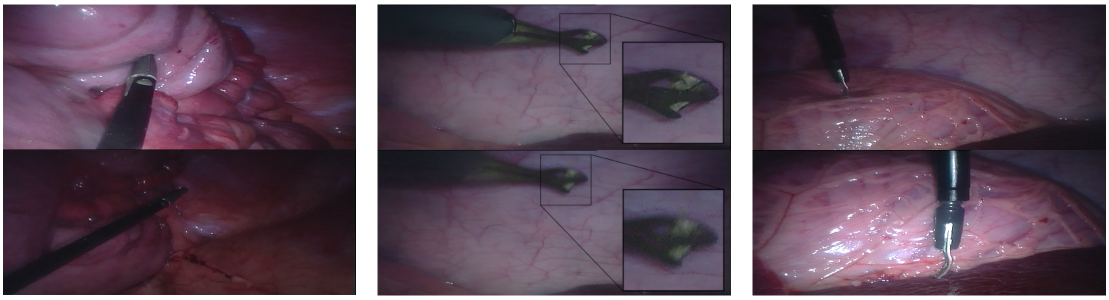
be processed individually with their prediction made without consideration for the values outside of the region's local neighborhood. In addition to this, the image data is often transformed into a representation that extracts only the most salient and discriminative parts of the signal, stripping away as much redundancy as possible. This processing is normally referred to as the construction of features and involves various linear or non-linear operations on the values of the pixels. These simple representations can then be used directly or alternatively accumulated into ensemble representations. In this section we will explore in detail the different types of representations the video data is normally transformed to when performing image based instrument detection in MIS.

The desired characteristics of features for object detection have the following criteria.

- **Uniqueness** The distribution of values taken by the feature should be as divergent as possible when comparing instances of different classes or parameter values.
- **Invariance to lighting changes** Ideally when the scene lighting changes and the illumination incident from the object surface is modified, the measured value recorded by the feature should be as close to static as possible. This can often be achieved by normalizing the values of the feature according to the mean illumination in the scene.
- **Invariance to viewpoint changes** The chosen features and the parameters required to correctly assign them to the various classes should not change as the camera or instruments are moved around the scene. Although this is a common property of color intensities at a particular pixel, spatially variant filter responses change regularly as objects rotate and move closer to or further away from the camera so in the case of these features, this is a desirable property.
- **Invariance to motion blur** Motion blur is a common problem in almost all video based computer vision tasks. Surgical instruments are often moved quickly across the camera field of view and the camera frame rate is often not high enough to avoid motion blur. Coarse groupings of pixel intensities are often not overly affected by this measure but features which rely on local intensity changes could be heavily affected by this type of noise.

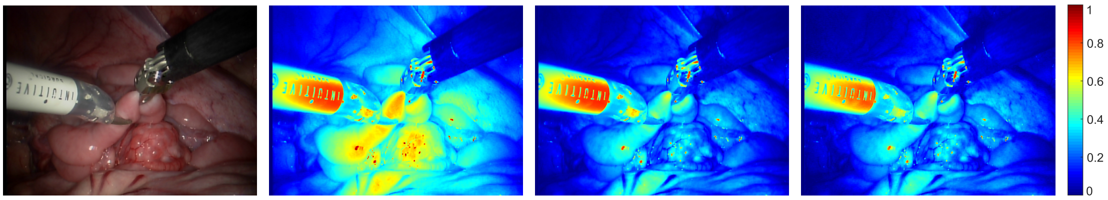
2.2.1 Color Intensity Features

The simplest type of feature which can be generated from a video signal is to transform the color intensity values into different color representations, which can in certain cases amplify visual differences between materials in the image. A color model is a mathematical representation for mapping tuples of numbers to colors which, when combined with a specific viewing interpretation results in a color space. Most color models are designed either to give good representation to the colors that can be seen by the human eye or alternatively to provide a system which enables a large number of different colors to be represented by as small a range of numbers as possible, which is advantageous for compression and data transmission [84].



(a) (b) (c)

Figure 2.2: (a) An example of how lighting conditions can make detecting an instrument more challenging. Rather than appearing highly discriminatively against the background, the dark appearance of the instrument shaft blends with the shadows. (b) The edges and detail on the instrument head are lost when fast motion occurs, however due to the high frame rates of modern cameras this is becoming less of a problem. (c) The instrument appearance changes as the viewpoint of the camera shifts. Features should ideally change minimally between these viewpoints to enable correspondences to be found.

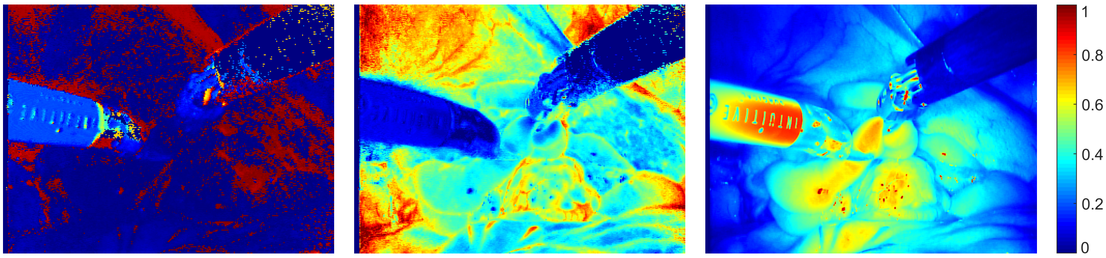


(a) RGG **(b)** Red **(c)** Green **(d)** Blue

Figure 2.3: An example conversion of a surgical image (a) into the RGB (b-d) colorspace. The images have been normalized to the range 0-1 and the single channel intensities are mapped to a more visually discriminative RGB representation.

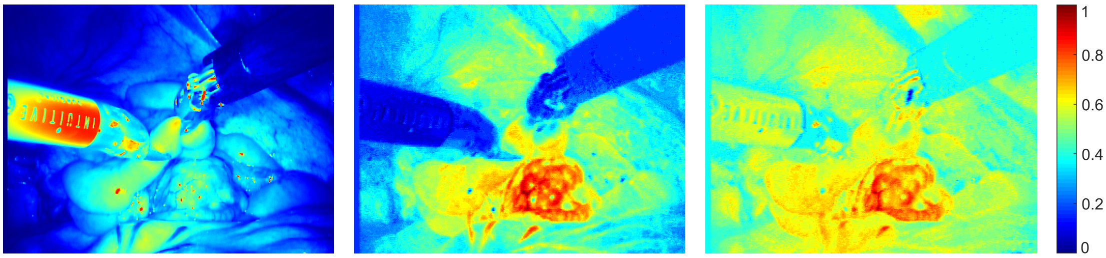
Red, Green and Blue (RGB) is by far the most common color model and is derived from the tristimulus theory of color which describes the human visual system as composed of three types of color sensitive cone cells in the retina, all of which have varying responses to photons across the wavelengths of visible light, with each type having a peak response in either the red, green or blue portion of the visible spectrum. Despite its common usage in computer vision segmentation work [85], it has had limited usage as a color intensity feature in medical imaging. Early applications [55, 59, 60] used small training sets to learn probability distributions over the RGB pixel intensities for an instrument and a tissue class. More recent applications [66] have used the red channel from the normalized RGB space which selects for pixels with red hues without being compromised by light or dark pixels which would have high red components without any perceptual redness. RGB is often used as part of patch based similarity measures, which derive their strength from one-to-one comparisons between pixels in two regions rather than the explicit representation of the image data. Robustness to lighting changes, which normally complicates the representation of color with RGB can be handled with zero mean sum-of-squared-distances (ZMSSD) [59] or sum-of-conditional variances (SCV) [73].

In more recent work [56, 58, 61, 66] the Hue, Saturation, Value/Luminance (HSV/HSL) colorspace has become very popular in the task of detecting instruments in MIS images. HSV is a conical colorspace which conveniently decouples luminance from chrominance. This is particularly useful as the represented colors are at least partially invariant to lighting changes. However, as the value component of this model is highly dependent on the ambient illumination level, it is often ignored [53, 57]. A particular challenge when comparing distances in the HSV colorspace is that typical norms, such as the L^2, are not particularly valid because HSV is a conical space and is therefore non-Euclidean. To address this, the ConeHSV representation which rescales the values to be Euclidean has been used as part of a detection



(a) Hue **(b)** Saturation **(c)** Value

Figure 2.4: An example conversion of a surgical image into the HSV colorspace. The images have been normalized to the range 0-1 and the single channel intensities are mapped to a more visually discriminative RGB representation.



(a) L **(b)** a **(c)** b

Figure 2.5: An example conversion of a surgical image into the CIE Lab colorspace. The images have been normalized to the range 0-1 and the single channel intensities are mapped to a more visually discriminative RGB representation.

system for robotic instruments [9]. Computing HSV from RBG is a simple operation summarized by the following relationships:

$$V = max(R, G, B) \tag{2.1}$$

$$V_{min} = min(R, G, B) \tag{2.2}$$

$$S = \frac{V - V_{min}}{|V|} \tag{2.3}$$

$$H = \begin{cases} \text{if } V = R & \frac{60*(G-B)}{V-V_{min}} \\ \text{else if } V = G & \frac{60*(B-R)}{R-R_{min}} + 120 \\ \text{else} & \frac{60*(R-G)}{V-V_{min}} + 240 \end{cases} \tag{2.4}$$

where the Hue values are shifted into a circular representation by 120 and 240 degree offsets.

CIE Lab is a colorspace based on an attempt to find a perceptually uniform color representation, which was a common limitation with the RGB and HSV color models [84]. This means that equal distances in the color space produce equal perceptive shifts in the observed color and enable different points in the color space to be compared with traditional distance metrics [86]. The 3 values of a CIE Lab color define lightness (L), difference between red and green (a) and the difference between yellow and blue (b) where the difference measures are known as opponent colors. These values provide a larger gamut of possible colors than RGB, which only uses narrow color bands, and better approximate human vision [87] but the consequence of this higher precision is that CIE Lab representations require a greater than 8-bit representation for each color channel, whereas RGB and HSV can be fairly well represented in the range 0-255. Its use in medical images is fairly limited however, CIE Lab colors have been used with other color based [80] and with color and gradient based cues [79, 82, 83] for both neurosurgical and laparoscopic surgical instrument detection. Conversion from RGB to CIE Lab involves a more complex process than HSV and additionally requires the selection of a white point to normalize the color values [87], where the standard choice and the value used in this thesis is the D65 illuminant which was designed to represent direct daylight. The other two whitepoints defined by the CIE 1931 standard are for daylight in shade and incandescent bulbs. To convert RGB to CIE Lab the intensity values must first be converted to an intermediary RGB representation, such as sRGB and Adobe RGB before being converted to CIE XYZ where the white point is assigned. The transformation is computed as:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.4124 & 0.3576 & 0.1805 \\ 0.2126 & 0.7152 & 0.0722 \\ 0.0193 & 0.1192 & 0.9505 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{2.5}$$

$$L = 116 f(Y/Y_n) - 16 \tag{2.6}$$

$$a = 500\big(f(X/X_n) - f(Y/Y_n)\big) \tag{2.7}$$

$$b = 200\big(f(Y/Y_n) - f(Z/Z_n)\big) \tag{2.8}$$

where $f(x)$ is given as:

$$f(x) = \begin{cases} x^{0.333} \text{ if } x > (\frac{6}{29})^3 \\ \frac{1}{3}(\frac{29}{6})^2 x + \frac{4}{29} \text{ otherwise} \end{cases} \tag{2.9}$$

where $X_n = 95.047$, $Y_n = 100$ and $Z_n = 108.883$ when using the D65 white point coefficients.

Dropping the color values and working directly with the grayscale intensity has had surprisingly promising results for a one-dimensional representation and has historically been used in automatic segmentation methods in computer vision [88, 89]. Using the grayscale intensity has been demonstrated in instrument segmentation as an effective feature to enhance HSV results by filtering for the instrument tips [75]. It is also been used as part of an adaptive representation of mutual information (MI) for tracking in retinal microsurgery [67], where illumination invariance is built in using a weighted joint intensity distribution. Computing grayscale from RGB is not as straightforward as taking the mean of the R, G and B channels when the desired outcome accommodates the varying sensitivity of the human eye to each primary color. Instead a weighted average of the three channels is taken:

$$G = \begin{bmatrix} 0.2126 & 0.7152 & 0.0722 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{2.10}$$

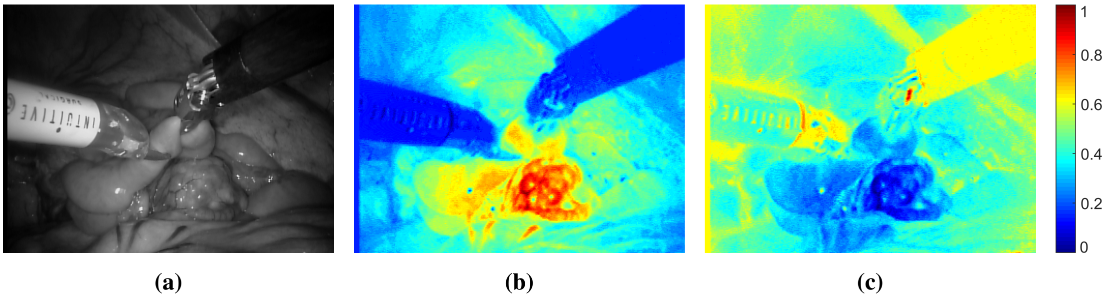
which is the Y component of the CIE XYZ color space.

Opponent color spaces, similarly to CIE Lab have their origins in human perception [90, 91] and are related to the differences between the red, green and blue color values. They have been used alongside other color models to detect instruments with random forest classifiers [79, 83]. They are computed from RGB as:

$$O1 = 0.5(R - G) \tag{2.11}$$

$$O2 = 0.5B - 0.25(R + G) \tag{2.12}$$

where $O1$ refers to the Opponent 1 color model and $O2$ refers to the Opponent 2 color model. As a color model which requires floating point values to represent the color values, images represented in this color space require more memory to process than 8-bit RGB, which can be computationally challenging for real-time applications.



(a) (b) (c)

Figure 2.6: (a) The grayscale response image. (b) The Opponent 1 color model. (c) The Opponent 2 color model. The Opponent images have been normalized to the range 0-1 and the single channel intensities are mapped to a more visually discriminative RGB representation.

2.2.2 Texture Features

The color features described in the previous section are all based on the intensity values at a single pixel and as such, do not include any neighborhood information in their descriptive power. Combining local regions into distributions of varying or constant intensities is known as texture and can be used to distinguish colorless, homogenous metallic instruments from tissue and other textured surfaces.

The most efficient and simple to compute are image gradients of which the simplest form uses central differences and requires just 3 pixels either vertically or horizontally adjacent to a given pixel. These are normally either computed directly on grayscale intensity values [63, 69, 83] but can in principal be estimated from individual channels of color models. Gradients can also be computed over a larger neighborhood through convoluting filters, such as the Prewitt or Sobel [92]. These kernels extend the central difference gradient computation with a smoothing operation which can either be uniform, in the case of the Prewitt operator or centrally weighted, in the case of the Sobel operator. Choosing a threshold for the gradient magnitude to select edges or other meaningful textual changes from image noise is particularly challenging and normally involves hand-tuning parameters, such as the classic Canny edge detector which has been applied to extract continuous contours around instruments [78]. Second-derivatives additionally provide information about the nature of zeros in the gradient image and have been used as part of filter banks alongside color and gradient based features [70].

Compared with the 3 pixel neighborhood used in gradient computations, larger spatial neighborhoods capture more information about the local texture. This is can be achieved with co-occurrence matrices whereby the intensity values in an $N \times M$ patch increment a matrix at element (i, j) every time the i^{th} and j^{th} intensities are adjacent to one another. As adjacency in a 2D plane can be defined as 4 possible configurations: left-right horizontal, up-down vertical and left-right and right-left diagonal, there are 4 different co-occurance matrices that can be computed for a given image which provides the features with some informal invariance to rotation. The original implementation of this feature type, known as Haralick Features [93] were based on grayscale intensity and they were used to compute up to 14 different statistical measures such as contrast, correlation, entropy and homogenity which have been used alongside color intensity features to segment robotic and retinal surgical instruments [9, 71]. Additionally, texture features such as Textons [94], which accumulate oriented sinusoidal filter responses into clusters, have been demonstrated for neurosurgical tools [82].

Gradient features which retain the entire structural integrity of the patch have difficulty when the image intensities change due to perspective transforms in the image. To counter this, spatial information can be discretized into histograms, which have been extensively demonstrated to provide significant invariance even in the case of large deformations. Histograms of Oriented Gradients (HoG) [95] have been popular in the computer vision literature, spurred mostly their success in object detection [96, 97].

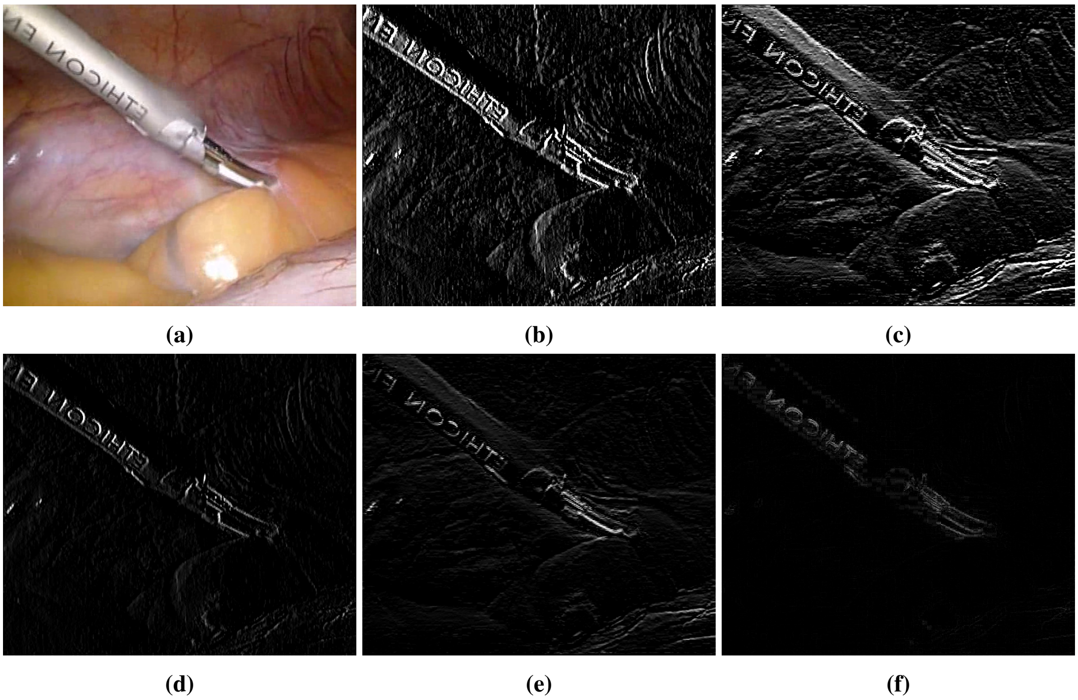


Figure 2.7: Images of MIS procedures captured through a laparoscope processed with different kernels. (a) The original image. (b) The Scharr x derivative kernel. (c) The Scharr y derivative kernel. (d) The Sobel x derivative kernel. (e) The Sobel y derivative kernel. (f) The Laplacian kernel.

These work by creating several histograms in rectangular blocks on a dense grid of uniformly spaced cells which are concatenated together to form a single feaure vector. Local contrast normalization is also used to improve the feature's robustness to lighting changes. In line with human and face detection applications, most users in instrument detection [74, 82] have focussed on constructing HoG features using upright rectangular blocks, which has limitations when the tracked object rotates in-plane. This has been countered in recent work [81] whereby a rotated coordinate system is created with a 2D tracking algorithm. Other histogram based texture features such as the scale invariant feature transform (SIFT) and speeded-up robust features (SURF) are common in computer vision [98, 99] and have been used as parts of locally restricted texture features [71, 70]. One significant disadvantage of accumulating large HoG is that they often have enormous computational cost, impacting the real-time potential of algorithms that rely on them. Faster implementations which maintain comparable accuracy such as Local Binary Patterns have been used to describe superpixel regions alongside multiple color features for surgical instrument detection [79].

2.2.3 Features from Multiple Images

An alternative method of computing features involves making use of images taken from different points in space or time. Disparity features are an example of using the spatial shift between the two camera images of a stereo laparoscope and compute the inverse depth of the scene at each pixel through a process known as 3D reconstruction [100]. This allows the fact that instruments are typically closer to the stereoscopic camera than the tissue surfaces in the body to be used as a detection cue [75]. This feature is of course reliant on the quality of the reconstruction algorithm which typically uses color or gradients to match correspondences but can be extremely useful due to the high-level smoothness constraints they contain. Motion cues exploit temporal measurements and measure the motion or flow of intensities around an image. This also provides a strong discriminative cue due to the distinctive motion patterns of the instruments. Fast motion causes problems when an interlaced video feed is used,

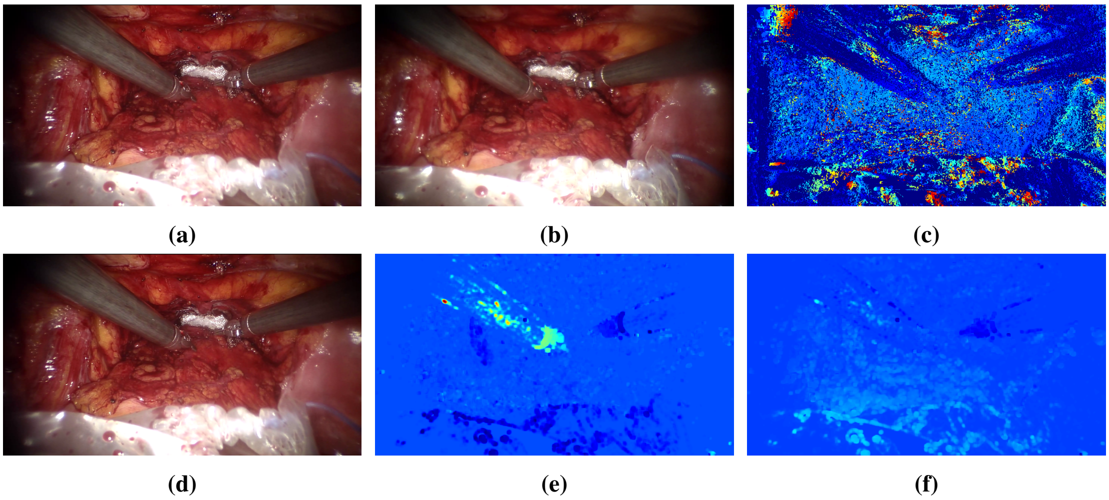

(a) (b) (c) (d) (e) (f)

Figure 2.8: (a) An example left camera eye image. (b) The corresponding right camera eye image. (c) An example disparity map computed with the Semi-Global Block Matching algorithm [5]. (d) The consecutive image captured by the left camera eye after (a). (e,f) The x and y dense optical flow fields [6] computed from (a) and (d).

so deinterlacing is often required to obtain cleaner results [75]. A particular challenge when working with motion features is discriminating from tissue motion caused by blood flow or respiration; this has been approached for laparoscopic images with thresholding [74] and generally in the computer vision field using techniques such as graph cuts [101].

2.2.4 Semantic Labelling

A popular higher-level feature is to use sementic labellings of the previously described features. This is achieved by using a parametric or non-parametric model of the lower-level features to divide them up into labels which represent distinct objects in the image. This process is often known as segmentation and provides a powerful method for detecting objects based on blobs [102]. Early techniques of estimating labels from features involves heuristic thresholding of values but as collected data sets grew larger in size statistical machine learning techniques began to be used to estimate model parameters which enabled accurate predictions to be made.

The simplest measure for assigning labels to non-parametric distributions over image data I is thresholding whereby a binary value is assigned to features if they all take on values larger than the defined threshold. The image data I can be transformed into any n dimensional feature representation for thresholding as:

$$I(\mathbf{x}) = \begin{cases} \text{if } \forall i(\mathbf{x}) \in I(\mathbf{x}) \; i(\mathbf{x}) > t_i & 1 \\ \text{else} & 0 \end{cases} \tag{2.13}$$

where $\mathbf{x}$ is a pixel location and we use $i(\mathbf{x})$ to denote a scalar value in the feature vector at $I(\mathbf{x})$ and the value 1 is assigned to a pixel if all n feature values in a exceed some threshold value $t = (t_1, t_2, ..., t_n)$. To provide multiclass thresholding, upper and lower bounds can be supplied rather than a single value. [56, 58, 80, 78] applied this technique to medical images, however, despite its advantages due to computation speed it has not been popular in modern methods due to it having no way of considering dependencies between variables and additionally it does not produce a probabilistic output, instead giving only a binary membership value.

Another traditional method is the Naïve Bayesian classifier which is considerably more popular than thresholding due to well defined parameter estimation and a probabilistic output. This makes the assumption that each dimension of the observed random variable (a pixel) is independently and identically distributed (IID) when conditioned on the labelling. Typically the probability density over the classes is chosen as a Gaussian distribution, which when combined with the IID assumption results in a tractable diagonalized covariance matrix. The class posterior is evaluated using Bayes rule where the prior is usually chosen as the probability $p(.)$ of an observation in the training set belonging to the target class. Given a joint distribution over a pixel location $\mathbf{x}$ in an image I, this is factorized as a product of distributions over individual features as:

$$p(c|I(\mathbf{x})) = \frac{p(c)p(I(\mathbf{x}))}{p(I(\mathbf{x}))} \tag{2.14}$$

$$p(c|I(\mathbf{x})) = \frac{p(c)\prod_{j=1}^{n} p(i_j(\mathbf{x})|c)}{p(I(\mathbf{x}))} \tag{2.15}$$

where Bayes rule is used to estimate a posterior distribution of the pixel label c, such as instrument or tissue, given the features at a particular pixel $I(\mathbf{x})$. The simplifying assumption of Naïve Bayes is that the features are conditionally independent given the label c, which allows Eq. 2.14 to be transformed to Eq. 2.15 where each dimension in $I(\mathbf{x}$ is denoted as $i(\mathbf{x})$. The advantage of this simplification is that a much smaller amount of training data is required to estimate the likelihood $p(I(\mathbf{x})|c)$ which simplifies to a one dimensional distribution and additionally for maximum likelihood (ML) solutions to the likelihood parameters can be computed in closed form [103]. Naïve Bayesian classifiers trained on the HS channels [57] or RGB data [55, 60] have been shown to be effective and fast classifiers for surgical instruments. A key component of the Bayesian classifier is the choice of prior distribution $p(c)$. Often this is chosen to model the frequency of instrument or tissue instances in the training dataset however a simpler alternative can be to use a unit prior in which case the Bayes classifier reduces to likelihood modelling [9, 71].

More recent models that have become popular in solving a wide variety of classification and regression computer vision problems such as keypoint recognition [104] and semantic image labelling [105] are Random Forests (RFs) [106]. They provide an accurate, fast and potentially parallelisable classification method and offer an easy extension to multi-class data, a useful feature for classifying multiple distinct tool or tissue types [7]. They effectively provide a similar decision structure to upper and lower bound thresholding, but provide a data-driven way for estimating the thresholds and increase robustness through ensemble voting and randomness. An RF is constructed as an ensemble of randomized decision trees, each of which consists of a set of weak learners that divides the classification of a sample $I(\mathbf{x})$ into a hierarchy of simpler problems. This is achieved by partitioning the sample space with decision boundaries and applying a different linear classifier in each region. Each applied classifier is either a decision node, which further partitions the search space and is represented as:

$$h(I(\mathbf{x}), \boldsymbol{\tau}) \in \{0, 1\} \tag{2.16}$$

where $h(.)$ is the hypothesis function, $\boldsymbol{\tau}$ is a parameter vector which dictates the shape and position of the partitioning j^{th} hyperplane, or a leaf node which labels the sample as belonging to one of the desired classes:

$$c^{\star} = \max_{c} p(c|I(\mathbf{x})) \tag{2.17}$$

where $c^{\star}$ is the labelling and $p(c|I(\mathbf{x}))$ is the posterior probability of the class c given the sample. They have been used recently in surgical instrument detection with different color and gradient type feature as part of a general surgical segmentation framework for making measurements of distances in gastric bypass procedures [79, 83].

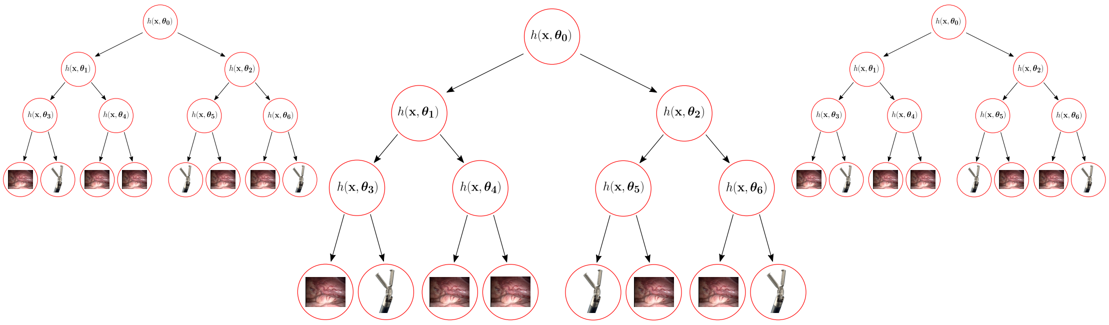


Figure 2.9: The RF model shows each tree consisting of red nodes where a single decision plane is applied to a sample directing it to one of two child nodes. Each decision plane is a linear classifier parameterized by $\mathbf{t}_{i,j}$ where i indexes the tree and j the node. After passing down the tree the sample arrives at a leaf node where it is assigned a label c, which in this case is either an instrument or tissue.

2.3 Connecting Features to Pose

Given a particular feature representation, pose estimation involves finding methods of computing the parameters that describe the object's pose directly from the feature representation. Most modern pose estimation techniques have taken a principled and probabilistic approach and are broadly divisible into two areas: generative or model based approaches and discriminative approaches although several older methodologies tended to forego a holistic modelling approach and instead solve the problem with multiple processing steps, which we will call algorithmic methods.

2.3.1 Generative Methods

Generative methods involve constructing a model of the image formation process whereby, given a specific parameter set, a representation of the image data can be generated. They are often described as model-fitting because the generative model is usually iteratively fitted to the image data. The advantage of these methods stems from the limited or non-existent requirement of training data compared with discriminative methods [107], and the ease with which they allow the incorporation of high level reasoning about the problem [108]. Their disadvantages come from their computational cost when seeking an alignment between the model and the data, particularly if the problem is high dimensional, as well as avoiding local solutions when using a gradient based optimization method.

When applied to surgical instrument pose estimation, the most popular generative model makes use of the shape of homogeneous regions or semantic segmentations as part of a silhouette or region matching framework. This type of method is popular in all areas of computer vision, primarily due to the speed of computation and robustness to noise that connected regions have. The generative model will predict a shape given some model configuration and then a shape matching scheme will be applied to sample from the space of possible shapes until a satisfactory example is found. Using the silhouette to predict pose is popular in specific pose estimation tasks in computer vision [109, 110, 85] particularly when tracking rigid or semi-rigid objects as the self-occlusion problem is more easily avoided. Generating the silhouette is often achieved with a standard rendering pipeline and a 3D model of the surgical instrument, however optimization has proved challenging with gradient-free methods proposed [9] that are slow to converge and regularly do not reach accurate solutions. Simpler models such as 3D cylinders [60, 64] and 2D parallelograms [55, 57] have been used but these methods are fundamentally limited as they cannot represent complexities in the shape that arise from articulation. They do however enable direct solutions through geometric methods that can be extremely fast, which is a common limitation of 3D rendering pipelines.

Alternatively to using regions to generate silhouettes, it is also possible to directly find edges in an image to which the target model is then aligned [111]. This has been popular in the computer vision literature particularly due to the reduction of the correspondence search to 1D which eased computational problems of 2D searches. In environments where the background is relatively clean and homogenous these methods have been demonstrated to be efficient and accurate for 2D and 3D pose estimation tasks however, the gradients recovered in medical images are normally hugely corrupted by texture and lighting variation on the tissue surfaces leading to intractable estimation [63]. Using the insertion point of the instrument, however, can be used to provide a directional constraint on the extracted gradients leading to much cleaner images [63, 69]. Modelling this insertion point is complex due to patient motion and has been achieved with external optical tracking systems [63] and with geodesic grids [69]. Normally the tip of the instrument is not included in the cylindrical model, which is only used to match to the sides of the instrument, and therefore a separate estimation phase is used to estimate the tip position. Otsu thresholding has been applied along the symmetry axis [63] which provided reasonable results in images where there was an unobstructed view of the instrument shaft with a 7 to 10 pixel distance between the estimated tip and the true instrument tip.

Another popular method of estimating the 2D or 3D pose of a surgical instrument is to minimize a distance metric between projections of a priori learned points that have a known location on the surface of the instrument and their matched correspondences in the image. With enough found correspondences this forms an overdetermined linear system of equations. This problem is commonly referred to in the computer vision literature as the perspective-n-point (PnP) problem and its solutions have been extensively studied [112, 113, 114, 115]. Typically the most successful features used in these systems are gradient based as they are much finer scaled than color features and as such allow more accurate localization. Using point based features to detect the pose of medical instruments made use of SIFT and HoG features learned around the head on da Vinci robotic instruments (see Figure 2.10) [7] and combined this with kinematic information from a robot to provide 3D pose estimation [70]. Although point matching can produce very accurate results in the case of uncluttered images, the reliance on the visibility of particular interest points results in serious challenges when trying to estimate pose in occluded environments.

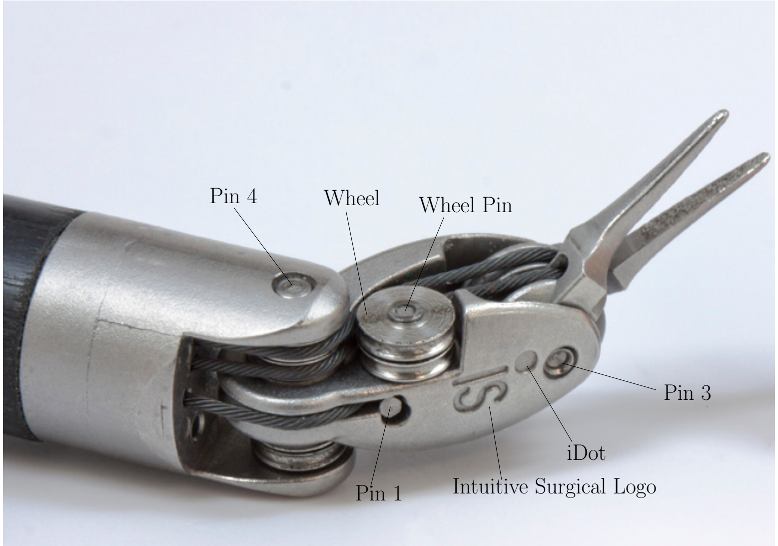


Figure 2.10: The detected features on a da Vinci large needle driver (LND) tool [7]. Image modified from ©2016 Intuitive Surgical, Inc. This instrument model is discussed in more detail in Chapter 6.

A final generative method of finding the pose of a surgical instrument is to use a 2D template representation of the instrument which is then deformably warped from the source image to a target image.

This typically works by finding the warp parameters that minimize a cost metric which measures the similarity between the warped patch and a region of the target image. These are popular techniques as they do not impose a prior model on the instrument appearance instead learning an updating representation online. 2D gradient template tracking has been demonstrated in retinal microsurgery using weighted mutual information to drive the warping [67] or alternatively using efficient second order minimization (ESM) [72] to make a coarse position estimate, before using the spatially weighted output of an adaboost trained cascade to get a final estimate of instrument position. The Line-Mod template matching method [116] has also been adapted for medical images [71] which creates a template from a CAD model offline at different articulations. At run-time, a subset of the templates are evaluated using the robot kinematics to define a reduced range of configurations. The same approach has been taken using a sum-of-squared-difference (SSD) template matcher [67].

2.3.2 Discriminative Methods

Discriminative methods on the other hand involve directly inferring the pose estimate from the configuration of image features, side-stepping the 2 stage modelling and inference process of generative methods. They make no high level assumptions about the nature of the function which performs this mapping and usually its form is learned directly from data. As labeled training data becomes more ubiquitous, these methods are increasing in popularity due to their low asymptotic error [107]. There has been limited introduction of discriminative methods to pose estimation of instruments (and other surgical vision tasks in general) despite these methods having recent success in solving computer vision pose estimation tasks [117, 118, 119] and primarily this has been due to challenges in obtaining realistic labeled training data in sufficient quantities. The majority of discriminative methods are much faster to evaluate than generative models so typically employ exhaustive search strategies rather than local sampling. However, due to their requirement on training data they often solve the estimation of 2D pose + scale as obtaining correctly labeled examples with pitch and yaw rotations is challenging for medical images. Normally the image x and y dimensions are searched in a sliding window manner where the detector is evaluated every n pixels. Additionally different rotations can be tested if the detector features are not rotationally invariant and different scales are searched by resampling the image.

The most traditional method of building a discriminative model for estimating instrument pose parameters is to assume a fixed template model which, although a considerable simplification, is much easier to train than more complex part based models [96]. A single part detector has been demonstrated for neurosurgical instruments [82] using a latent SVM trained on HoG features with a global shape constraint. The same authors additionally trained a Random Forest on 10 different feature channels but the runtime evaluation was prohibitively slow.

Several methods however have attempted to tackle the deformations in appearance due to articulation and out of plane rotation. The naïve approach would be to collect more training data for each of these examples but this increases the runtime and also the time taken to acquire training data. An approach to handle this type of problem that has been very popular in human detection in computer vision is to model the appearance as a spatial arrangement of parts whereby each part is detected separately and a simple distribution is learned to model the relative orientation of two parts, which has been used for tracking surgical instruments in video [72]. Latent support vector machine (SVM) has been used in combination with HoG features and a star model [120] to detect instruments [74]. As well as SVM, RFs have been demonstrated for articulated tracking in retinal microsurgery [81]. A single window detector is used to estimate a bounding box around a retinal instrument and HoG features for several deformable parts are independently detected within this box. This enables detection of the articulated clasper of a retinal instrument at frame rates of up to 30 Hz. The particular advantage of discriminative methods such as SVM or RFs is that assumptions about the object appearance are not enforced by the designer, they are instead learned directly from the data. This means that they are less likely to contain simplifications or bias that reduce the accuracy.

2.3.3 Algorithmic Methods

Some methods do not strictly fall under the umbrella of generative or discriminative methods as they do not attempt to model the pose estimation holistically, instead treating it as a set of independent processing steps. Semantic label images are often used as a first step [50, 66] and these are usually searched with Hough transforms to find lines and edges that can be attributed to cylindrical models. Many of these methods suffer from noise around the semantic segmentation where mislabeled pixels create breaks or islands in the label image which can be solved with erosion and dilation filters [62]. Once outliers have been removed, a single connected region needs to be found which can be achieved with region growing [58, 61] where the seeds are initialized in the largest blobs and extend until a predefined intensity gradient threshold is found. Given a single connected component, its shape can be processed to estimate pose using moment of inertia analysis [62] or Hu's moment [58] which potentially have greater robustness to small amounts of noise around the edges which may disrupt straight line extraction. Rather than shape analysis, the edge or center lines can be estimated [78] which enables a 1D search line for the instrument tip which appears as a maxima in the gradient. Alternatively, spatial constraints such as the maximal distance from the center of mass of the region can be used [79]. As the estimates of orientation are often quite noisy, this parameter can be ignored and a sole 2D coordinate can be tracked as the center of mass of segmented regions [56, 50]. A major limitation of algorithmic methods is that they require parameter tuning for each stage in different datasets. The active testing approach provides a method of combining several distinct discriminative models algorithmically enabling the estimation of each parameter as part of an entropy minimization framework. Each phase in the estimation extracts information from the previous phase informing its search space and enabling efficient and accurate solutions for 2D pose estimation in retinal microsurgery [68].

2.4 Temporal Tracking

The final component of a visual tracking method is to combine frame-to-frame measurements to obtain an estimate of pose over an extended sequence. There are two main fields of thought in this area: tracking by detection which treats each frame as independent from the last and temporal tracking where information from prior states is able to influence the predicted state for a given frame. Tracking by detection has come into prominence in recent years as computational improvements have rendered it tractable to re-estimate the entire object configuration at each frame when performing 2D tracking. However, when working in 3D or when facing challenging visual data due to occlusions or complex object configurations, temporal tracking is required which makes use of first and second derivatives of position as well a motion model to produce estimates of the likely configuration in a subsequent frame. This enables the search space for subsequent frames to be greatly reduced. Any method which makes use of information from prior frames suffers from problems of drift, when errors in the parameter estimate begin to accumulate and corrupt later measurements.

The most popular way of combining a motion model with the information from prior frames is with a Kalman filter [121]. The role of the Kalman filter is to predict a distribution over the pose parameters $\boldsymbol{\theta}$ at time t given a set of measurements and a set of previous values for $\boldsymbol{\theta}$. This is achieved by first computing a prior distribution for $\boldsymbol{\theta}$ given all of the measurements up to the previous timestep $t-1$ as:

$$p(\boldsymbol{\theta}_t|I_1(\mathbf{x}),...,I_{t-1}(\mathbf{x})) = \int p(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1})p(\boldsymbol{\theta}_{t-1}|I_1(\mathbf{x}),...,I_{t-1}(\mathbf{x}))d\boldsymbol{\theta}_{t-1} \tag{2.18}$$

where $I_t(\mathbf{x})$ reflects the image data acquired at t. This is combined with a probability distribution over

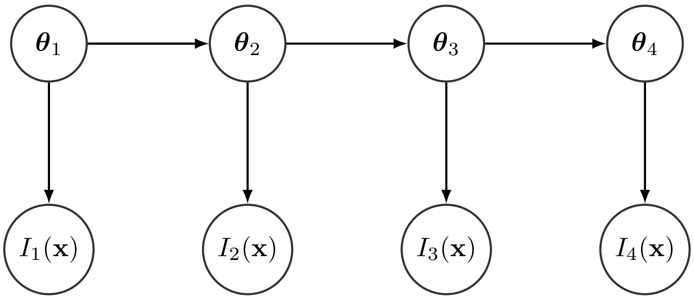


Figure 2.11: A Kalman filter allows estimates of the state at a time i, $\boldsymbol{\theta}_i$, to be estimated from prior estimates $\boldsymbol{\theta}_{i-1}$ and measurements $I_i(\mathbf{x})$.

the current measurement at t, $p(I_t(\mathbf{x})|\boldsymbol{\theta}_t))$ using Bayes Rule to form a an estimate for the current state as:

$$p(\boldsymbol{\theta}_t|I_1(\mathbf{x}),...I_t(\mathbf{x})) = \frac{p(I_t(\mathbf{x})|\boldsymbol{\theta}_t)p(\boldsymbol{\theta}_t|I_1(\mathbf{x}),...,I_{t-1}(\mathbf{x}))}{p(I_1(\mathbf{x}),...,I_{t-1}(\mathbf{x}))} \tag{2.19}$$

The Kalman filter makes the assumption that the dynamical models are linear and that $p(\boldsymbol{\theta}_{i+1}|\boldsymbol{\theta}_i) \sim \mathcal{N}(\boldsymbol{\theta}_i, \sigma^2_{i+1})$ and that $p(I_i(\mathbf{x})|\boldsymbol{\theta}_i, I_{i-1}(\mathbf{x})) \sim \mathcal{N}(\boldsymbol{\theta}_i, \tau_i^2)$ [108], where $\mathcal{N}$ refers to the Gaussian distribution and σ^2_{i+1} refers to the process covariance at the updated timestep $i+1$ and τ_i^2 is the measurement covariance at i. This model is particularly popular because it provides optimal predictions if the assumptions are valid. A linear Kalman filter has been used to combine measurements from a da Vinci robotic control system with visual measurements as part of a robotic servoing system [59] which can help to mitigate errors that occur due to visual occlusion. To avoid failed visual observations from corrupting their state estimate, they threshold their visual observation confidence and only make use of the measurement prediction from the Kalman filter when the threshold inaccuracy is exceeded. To address limitations in the linear motion model and enable a polar coordinate representation which better represents the constraints of a single insertion point, the extended Kalman filter (EKF) can be used [78] which allows for non-linear functions to be used as the motion model. A special case simplification of the recursive Bayesian filter is to use an identity matrix as the motion model, which causes the estimate in the new frame to match the estimate from the previous frame [55, 67, 72, 73, 74, 76, 80, 81]. We refer to this technique as tracking-by-initialization.

A considerable difficulty with the aforementioned Kalman filtering approaches is that they represent the probability distribution over the state as a unimodal normal distribution. Although this is often accurate, there are many situations where a multimodal distribution better approximates the true distribution due to there being many competing alternatives for the world state. Particle filters represent the probability function over the state with a set of particles which are evolved through time by a particular model of the system transition. A well known particle filtering method is the Condensation algorithm [122]. Each particle represents one estimate of the system state and at each timestep it is projected through a possibly non-linear state transition function giving a new estimate of the system state. This estimate is then evaluated giving a probability of its accuracy. A new set of particles can then be estimated by resampling from this new distribution. The Condensation algorithm is popular in surgical instrument tracking [57, 60, 50, 75] due to its ability to track through the multiple occlusions faced in surgical environments.

2.5 Conclusion

This review covers the main contributions to pose estimation in the literature. As the field is quite new, there is limited consensus on exactly which methods work best and typically due to a lack of an accepted validation methodology and open data it is difficult to compare different methods effectively. However, when aiming to recover 3D pose directly from images, the majority of methods have focussed on region segmentation and in particular using 3D rendered models [9] showed particular promise as it can leverage the full shape information and allows easy extension to different instrument types and camera views. However, limitations have centered on inaccuracies in the region segmentation relied upon to estimate the pose causing difficulty in making reliable predictions. Furthermore, the lack of differentiable energy functions have lead to gradient free approaches [9] which are slow to converge and unreliable at reaching local or global optima within a reasonable time limit for high degree of freedom problems. These observations shall direct the remaining body of this thesis as we will aim to first produce highly accurate region segmentations before moving on to creating a differentiable pose estimation framework that will allow us to estimate the 3D pose of articulated instruments. We aim to achieve this by using principled generative modeling techniques which will allow our techniques to be extended and improved without significant modification.

The main challenges that our method will have to be able to handle include highlights and specular reflections on both the instrument and the tissue in the background which are difficult to disambiguate from one another and are often confused with metallic surfaces. Another challenge will be dealing with the instrument routinely moving in and out of the field of view, which occurs regularly in surgical procedures. This will require a detection method to determine when the instrument is out of view and a fast and reliable reinitialization for when it returns to the image. A final challenge will involve dealing with instrument appearance changes, such as blood, which can make detecting features particularly challenging on surgical instruments.

Chapter 3

Semantic Segmentation of Surgical Instruments

3.1 Introduction

The problem we aim to solve in this chapter is how to correctly segment instruments in surgical images. Segmentation is a common problem in computer vision and is often simplified to a foreground-background division [123] whereby the object of interest is represented by a single class. More powerful models have been recently developed and a common trend is to construct multiple classes for different semantic regions of an image [124, 97, 125]. The problem is distinct from 2D bounding box multi-object detection, which is popular when the object of interest, such as a human body or a face, is normally in some standard ‘upright’ orientation and can be reasonably represented by a simple, convex shape. However, when the object of interest either exhibits significant in-plane rotation or has a more complex shape that needs to be captured, a single bounding box is not sufficient. Either complex hierarchies of multiple rotated bounding boxes [96] or direct pixel labelling is needed to capture the complex shape. The advantages of semantic segmentation is that complex objects can be represented and their limits within the image precisely defined which enables more complex inference tasks to be performed using their shape or size.

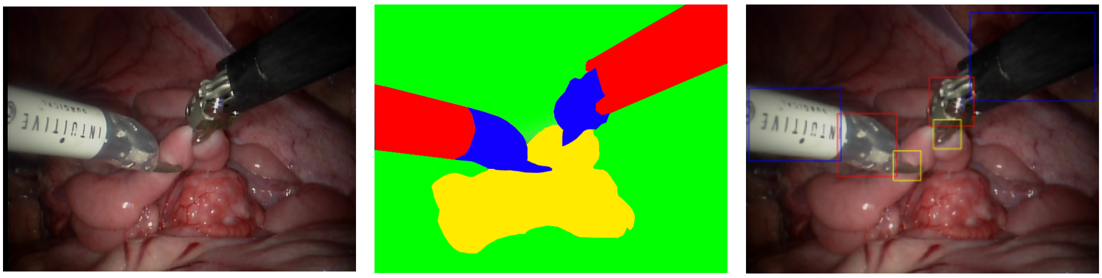
The basic idea behind semantic object segmentation is to process a 2D image signal and by extracting local or global features from neighborhoods in the image, produce an output image where each pixel is labeled as belonging to one of a set of classes (see Figure 3.1b). The first stage in engineering a solution to a segmentation problem is to select features in the signal that will be used to make the class predictions and then this is followed by the selection of a model which will use these features to make predictions. Semantic segmentation can be formulated mathematically by first transforming the image into a more descriptive representation as:

$$\forall \mathbf{x} \in \Omega \quad g : \hat{I}(\mathbf{x}) \mapsto I(\mathbf{x}) \tag{3.1}$$

where the image $\hat{I}$ is the raw pixel data captured by the surgical camera which is transformed by g at pixel locations $\mathbf{x}$ into the feature representation $I(\mathbf{x})$ over the domain of the image Ω. The elements of I are then passed into a predictive function f to generate the individual class labellings c:

$$c = f(I(\mathbf{x}), \chi) \tag{3.2}$$

where χ represents the parameters of the predictive model. The analysis of features and learning methods for semantic segmentation in section 2.2.4 demonstrated that there is little consensus amongst the community on the best feature representation and methods for semantic segmentation of instruments in



(a) **(b)** **(c)**

Figure 3.1: (a) An example image captured during a robotic surgical procedure. (b) An example of how the image can be segmented into different regions where red pixels represent the instrument shaft, blue pixels represent the instrument's articulated wrist, yellow pixels represent an anatomical object being manipulated and green represents the background. Note that this segmentation is performed by hand and is not the result of image processing. (c) An example bounding box detection where different regions of the instrument are surrounded by a single colored box.

surgical images. In this chapter, we will seek to determine a set of features which can be applied successfully for instrument segmentation with a well developed and popular learning method, RFs. Although in an ideal sense, different learning methods would be additionally experimented with, the time frame required for a thorough analysis of these methods is beyond the scope of this thesis. The work presented in this chapter makes up part of the publication [126]. Since this publication, recent work in semantic segmentation with convolution neural networks (CNN) [127, 125] have demonstrated excellent results and will likely surpass the presented results. However, adapting CNNs to the limited training data of instrument segmentation is non-trivial and is an open research problem.

3.2 Feature Evaluation and Segmentation with Random Forests

Our objective for this chapter is to determine which combination of features produces the most accurate segmentation of surgical instruments balanced against the processing time required to compute the features and then classify them. We use RFs as the classifier for our experiments due to the availability of numerous open source implementations [128, 129], their flexibility on number of samples and dimensionality of input variables, the fact that they possess an internal method of computing training accuracy and ranking variable importance and that at the time of this work, they produced state-of-the-art results on numerous datasets. These properties make it easy to experiment with RFs and the variable importance ranking provides a built in method of assessing the suitability and strength of different features. RFs were introduced in Chapter 2 and here we will present how they were implemented in this thesis and how they are trained.

3.2.1 Training a Random Forest

RFs are trained with supervised learning which entails collecting a training dataset $\{\mathcal{X}, \mathcal{Y}\}$ where training samples $I(\mathbf{x}) \in \mathcal{X}$ are assigned to manually labeled ground truth $\mathbf{y} \in \mathcal{Y}$. We are working on classifying image pixels so the training data used is numerical but bounded by the precision of the datatype used in storage and the selected labels are categorical with one representing the background, one representing the instrument shaft and one representing the instrument clasper. We train our forest using bootstrap aggregating or bagging [106] which increases the generalization of the resultant classifier by only training each tree on a subset of the data generated by uniform sampling with replacement. This is a method by which randomness is added to the forest because each single tree is trained on a slightly different set of data thus resulting in a different structure. Each tree is grown incrementally by creating a new node, then choosing a splitting parameter vector $\boldsymbol{\theta}_j$ which maximizes an information gain type metric. This provides a further method of injecting randomness to the tree, as it is possible to randomly select a subset of all possible splitting parameters and maximize over these instead of the full set:

$$\boldsymbol{\alpha}_j^\star = \max_{\boldsymbol{\alpha}_j \in \mathcal{T}_j} I_j \tag{3.3}$$

where $\mathcal{T}_j$ is a subset of the possible parameter vectors and I_j is the information gain type metric. Although it is possible to use linear combinations of the features as splitting functions, the most common technique is to use axis-aligned splits which divide the data along one feature value [130]. Although entropy [131] is a popular measure of information gain, the Gini coefficient is the classic measure [106] to compute the best split of the training data and is the measure used in this thesis. It is computed as:

$$\Delta i(s,t) = i(t) - p_L i(t_L) - p_R i(t_R) \tag{3.4}$$

where the impurity decrease of using the split s on node t is given by the impurity $i(.)$ at node t before the split subtracted by the weighted impurities of the left t_L and right t_R nodes. The impurity $i(.)$ is given by the probability that a randomly chosen element in the node would be misclassified if it were randomly labeled using the distribution of labels in the node and is defined as $\sum_{j \neq k}^{C} p_j p_k$. The weights are given by the number of samples passing to the left and right nodes respectively:

$$p_L = \frac{N_L}{N} \tag{3.5}$$

$$p_R = \frac{N_R}{N} \tag{3.6}$$

where N_L and N_R are the number of samples that end up in the left and right nodes respectively. N is the number of samples in the parent node. Splitting is ceased when the number of training samples in a node is less that a predefined number which we set to 0.1 $\%$ of the training data set size.

There are two normal criteria for deciding when to cease training, either when sufficient accuracy has been reached at classifying what are known as out-of-bag samples, which are the samples from the training set which are not included in the tree's bagged dataset, or alternatively when the number of trees in the forest has reached a predefined threshold. As the classification time increases approximately linearly with the number of trees in the forest, increasing the number of trees in the forest for only a small increase in the accuracy may not make sense in the context of time critical systems. With this limitation in mind, we cease training on a fixed number of trees and vary this fixed limit over a number of experiments to determine when the performance increase begins to plateau as new trees are added.

Many machine learning methods require feature normalization before training which commonly involves modifying the data such that it has zero mean and unit standard deviation. The effect of feature normalization is that it allows different features to be fairly compared to one another enabling distances to be computed in the feature space. For example, if samples of one variable can have much larger differences than samples of another, this variable will invariably be more heavily weighted when computing a metric such as Euclidean distance. However, when training a RF using axis-aligned features, feature comparisons for a proposed split are only computed on a single feature so the relative sizes become irrelevant.

3.2.2 Classification with a Random Forest

Run-time classification with a RF is performed by individually assessing the input sample independently with each tree in the forest. The sample is passed from the root node, down to the left or right child node of a single node in each level of the tree until it reaches a leaf node where it is classified. Each leaf node contains the most common label in the remaining samples in that node during training and this is

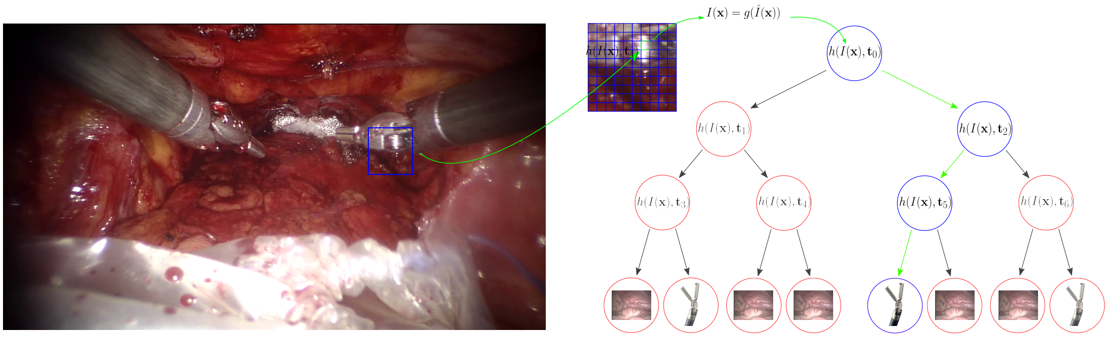


Figure 3.2: Classification with the random forest model is achieved by processing each pixel into a feature representation which is then evaluated by nodes in the tree. For the sample in this example, it passes into the root node, from which it is directed to node 2, then to node 5 and from there it is classified as an instrument. Its path is shown in blue.

assigned as the tree's vote for the sample's class label. This process is repeated for each tree and the final label is assigned in a winner-takes-all strategy. Some implementations [129] enable a weak-probabilistic output where the class posterior probability is given by the fraction of trees that voted for the class.

3.2.3 Feature Ranking with Random Forests

One of the most significant advantages of using RFs for detection is that they provide a built-in method of assessing the strength of the different features used in training. This inbuilt ranking is known as variable importance and uses the impurity decrease for all nodes where the feature of interest p in the feature vector $I(\mathbf{x})$. This is defined as:

$$Imp(I(\mathbf{x}), p) = \frac{1}{N_T} \sum_{T \in N_T} \sum_{t \in T : v(s_t) = x_m} p(t) \Delta i(s_t, t) \tag{3.7}$$

where t is a node in the tree T in the forest N_T. $v(s_t)$ is a function that returns the variable used in split s on node t and $\Delta i(.)$ is the impurity decrease of Equation 3.4. This measure is known as the mean decrease in impurity and gives a score for each tree which is then averaged over all of the trees in the forest. It has been shown to be a reliable measure for assessing how useful a feature is [132].

3.2.4 Analysed Features

We explore different color and texture features as the most straightforward method of evaluating which features provide the best predictive strength for surgical instruments. We use RGB, HSV, CIE-Lab and Opponent 1 and 2 as color features, which have been common in medical instrument detection [66, 61, 9], but also experimented with Gabor filters as a texture feature representation. The first stage in constructing the features using the equations laid out in section 2.2.1 is to acquire images in the RGB colorspace, from which all other colorspaces were computed. Images from surgical cameras on systems such as da Vinci are transmitted over a SDI connection and are encoded in 8 bit YCbCr colorspace which divides the pixel information into luminance (Y), blue difference chrominance (Cb) and red difference chrominance (Cr). The advantage of this encoding over standard RGB is that the RGB color model contain significant redundancy as small differences in 2 RGB values can have little perceptual difference for the human eye. To exploit this and reduce bandwidth, YCbCr separates luminance and chrominance and transmits the luminance at higher resolution than the chrominance, a technique known as chroma subsampling. As the human eye is less sensitive to small chrominance differences than luminance difference this effectively maintains visual quality at a reduction in bandwidth of up to 25 $\%$. The da Vinci transmits visual information at a chromanace subsampling of 4:4:2 (see Figure 3.3).

The transformation from YCbCr back to RGB is a trivial linear operation using ITU-R Recommen-

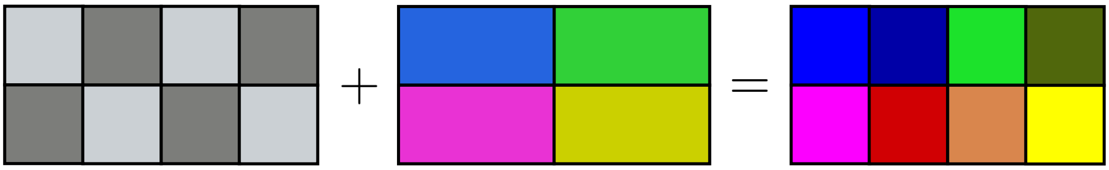
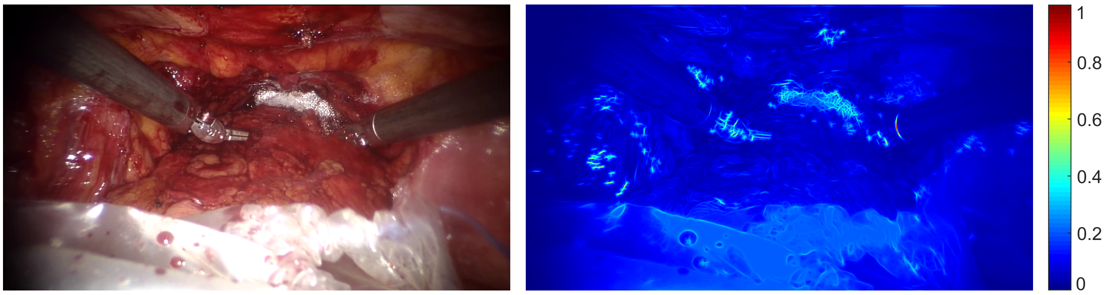


Figure 3.3: The 4:4:2 YCbCr macropixel. The luminance data (left) is sampled at full resolution but the chrominance (center) is sampled at half the frequency. These are combined together when creating the final image (right).

dation 709 conversion matrix:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 1.5748 & 0 \\ 1 & -0.4681 & -0.1873 \\ 1 & 0 & 1.8556 \end{bmatrix} \begin{bmatrix} Y \\ Cr \\ Cb \end{bmatrix} \tag{3.8}$$

Using this RGB color data, we compute HSV, CIE-Lab and Opponent 1 and 2 colorspaces and additionally compute texture features in the form of Gabor filter features. Gabor filters are used in many edge detection tasks and work by convoluting a Gabor kernel with an image. The Gabor kernel is a biologically motivated model which has a strong connection with the mammalian visual cortex. The filter is constructed as a Gaussian kernel modulated by a sinusoidal plane wave. A filter bank of differently oriented filters are created by rotating the angle of the sinusoid and by selecting the maximum response at each angle, an orientation invariant edge detector is produced.



(a) **(b)**

Figure 3.4: (a) An example image from a surgical procedure. (b) The extracted Gabor filter output which has been normalized to the range 0-1 and the single channel intensities are mapped to a more visually discriminative RGB representation.

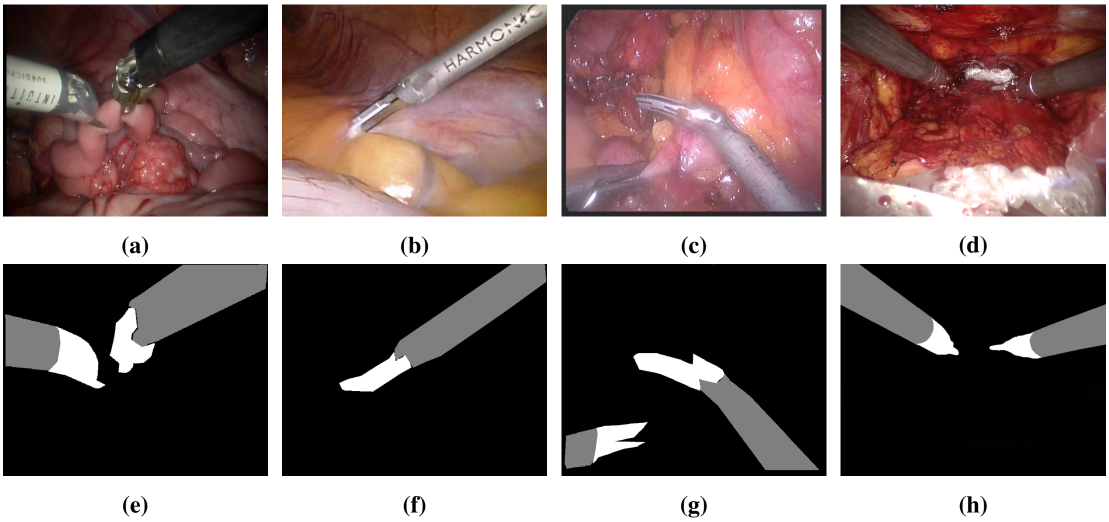
3.2.5 Dataset Construction

A dataset of 112 images from 7 different procedures showing different instrument types and tissue backgrounds were manually segmented, example frames from which are shown in Figure 3.5. We label the shaft of the instrument, the metal clasper and the background to create 3 distinct classes and each class is represented in the hand labeled images with a different color.

We use recursive feature elimination to select the optimal set of features for our random forest. This is a general feature selection process whereby the estimator is trained on an initial set of features and, using weights that can be assigned to each feature from a round of training, the worst performing features are pruned from the set and training is repeated. This method was originally popular with SVMs [133] but has recently been applied to RFs [134, 135].

3.3 Experiments and Results

Through our experiments we wish to answer 2 key questions. Firstly, we wish to understand which set of features provide the best performance on the forest when balanced against a requirement of reasonable


(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 3.5: (a)-(d) Example frames from 4 of the 7 datasets. The images vary in resolution between 720 $\times$ 576 and 1920 $\times$ 1080 and are stored in the RGB colorspace. (e)-(h) Example ground truth images for the datasets. Black pixels represent the patient tissue and any other background objects, gray pixels represent the instrument shafts and white pixels represent the instrument claspers.

classification time. For surgical images, due to the complexity of light reflectance in the scene, it is necessary to provide a thorough examination of different descriptions of the pixel data observed in each image to determine how much of this data is needed and how much can be discarded. In the following section we explore several different features including different colour spaces and texture features to determine which provides the most discrimination between the instrument and the tissue in our datasets. The second question we wish to answer is whether the most effective approach for detection and tracking systems is to train a general RF classifier offline on a large and varied set of training examples with the objective that it will be able to generalize over new examples or alternatively whether it is more reasonable to train a specific RF classifier on a minimal set of training data, such as the first frame, for each individual evaluation case. In our experiments we use the OpenCV CPU implementation of RFs [129] where we leave the majority of the parameters of the forest to their default values. The only exceptions are that we vary the number of trees in the forests from 1 tree to forests of 3, 5 and 10 trees. Additionally, we use a minimum sample count of 1 $\%$ of our training data set size and we allow the forest to choose from the entire feature vector when choosing the ideal splitting function.

To assess the accuracy of the features in detecting our 3 classes, we use two standard classifier accuracy measures: precision and recall [136]. Precision of class i is defined as:

$$P_i = \frac{TP_i}{TP_i + FP_i} \tag{3.9}$$

and recall of class i as:

$$R_i = \frac{TP_i}{TP_i + FN_i} \tag{3.10}$$

where TP_i is the number of true positives, which indicate a correct classification of a sample as a particular label i. FP_i is the number of false positives , which indicate an incorrect classification of a sample with the label i and FN_i is the number of false negatives is the number of instances where the classifier failed to correctly classify an instance of the class i. This leads the precision measure to indicate for a given class i the probability that if it assigned a label i to a sample, then it was correct. As this fails to account for missed examples (e.g. false negatives), the recall measure indicates the probability that this classifier will correctly identify instances of class i. To provide a more qualitative understanding of the effects of precision and recall on a classifier we can consider the type of output we would expect to see. Precision and recall are related metrics when we have a finite set of labels or no *unknown* class, for example, false positives for one class (lower precision) must result in false negatives (lower recall) for another class. A classifier which achieved high precision for the clasper and shaft labels and lower recall for the background class would correctly segment all of the pixels belonging to the instrument but with the possibility of connected blobs or regions which should have been labeled as background. In the alternative case of a high recall classifier for the clasper and shaft with lower precision for the background we would see very few blobs or regions that have been mistakenly classified as instrument, instead seeing holes in the instrument body which have been mistakenly classified as background.

The precision and recall scores for the RF are useful for indicating how well a particular RF is performing with a given set of features. However, we also require a way of understanding which features can be removed from the evaluation while retaining good levels of performance. Rather than removing features at random and assessing precision and recall, we can work more efficiently by removing features which give weaker predictive strength in rounds until we see a noticeable performance drop off. We achieve this by training on each set of $N - 1$ datasets and accumulating the variable importance scores from each round until we have completed the set of N leave-one-out training and testing phases. We eliminate the worst performing feature, as described in section 3.2.5, and increase the score for each feature if it was included in a round of training. More popular features will be included in more training rounds during recursive feature elimination and thus have a higher score. Inter-round variable importances cannot be compared directly as their magnitude is only relevant for comparing features within a single training round.

3.3.1 Multiple Dataset Evaluation

In this experiment, we attempt to assess the ability of the forest to generalize over new sequences. To achieve this we perform a leave-one-out evaluation over all 7 sequences in our dataset. One dataset is selected as a testing set and the remaining datasets are chosen as training sets. Variable importances for each feature in $\mathcal{X}$ are estimated using all the images in the 7 datasets. Evaluation is performed using the testing set and scores are recorded for the three object classes we are interested in classifying. We average the score across all of the datasets to obtain a single score for each training round. We remove the worst feature and repeat the process until we have only 1 feature left. This process is repeated for forests containing 1, 3, 5 and 10 trees. We demonstrate the accuracy of the forests in plots of the numerical results in Figure 3.6.

The results show that there is little improvement by adding more features and by adding more trees to the forest, suggesting that the limitation is more fundamentally related to the ease which the pixels in surgical images can be classified without consideration of neighborhood information and prior knowledge to avoid ambiguities. The precision of the background class is typically high which is likely due to the difficult or ambiguous regions being small relative to the background class as a whole. When performing class balancing the frequency of these ambiguous background samples then becomes small relative to the total number of instrument samples meaning the forest is overwhelmingly more likely to classify these samples as instrument rather than background. A potential solution to this would be to not perform class balancing but as we have the long term goal of using the segmented regions to estimate pose, high instrument recall is more important than precision. There is limited performance loss from removing features across all classes until 3 or 4 remain and then performance declines suddenly. The most likely explanation for this behaviour is the high degree of redundancy in the color features which renders many of them effectively useless in providing good classification. We show example classification images in Figure 3.8, where for brevity we show the images from the 3 tree classifier as there was not significant difference between the cases when the number of trees was varied. In these

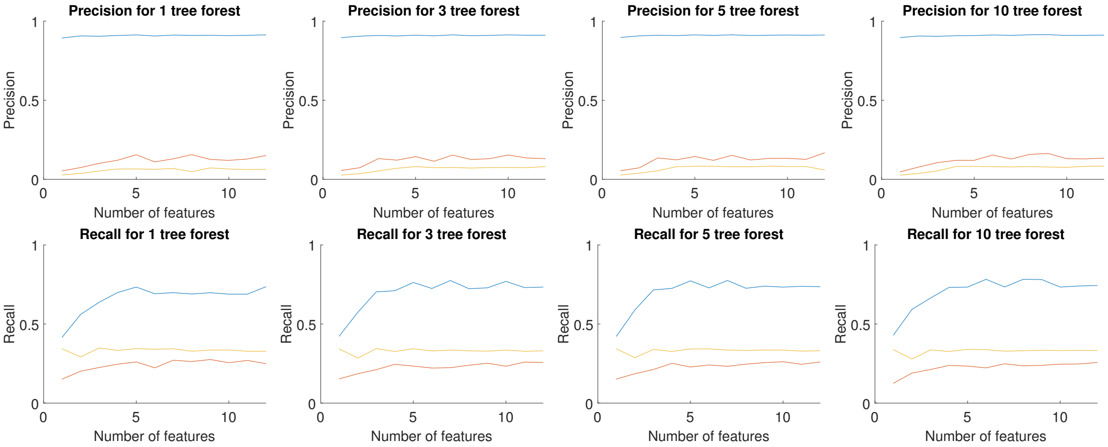


Figure 3.6: Background, Shaft, Head. The precision and recall curves for the 3 target classes when training 1, 3, 5, and 10 tree forests using multiple datasets. The values given are the average precision and recall across all 7 datasets when training on the data from $N - 1$ datasets and evaluating on the remaining dataset. The precision and recall plots when varying the number of trees show almost no change suggesting that the challenge lies much more in the feature strength than the classifier complexity.

images, it is clear that the redder tissues are more easily distinguished from the instruments but as the variance in the tissue appearance is high, a leave-one-out approach does not enable good generalization particularly with the lighter tissue examples which are regularly misclassified as instrument clasper.

The feature importances shown in Figure 3.7 are consistent across forests of all sizes and generally O1, Gabor and a and b from the CIE Lab space are the most used features. Given that the Gabor filter response is the only feature constructed from texture it is unsurprising that it is selected often by the RF, as the information that it provides cannot be replaced by any of the color features.

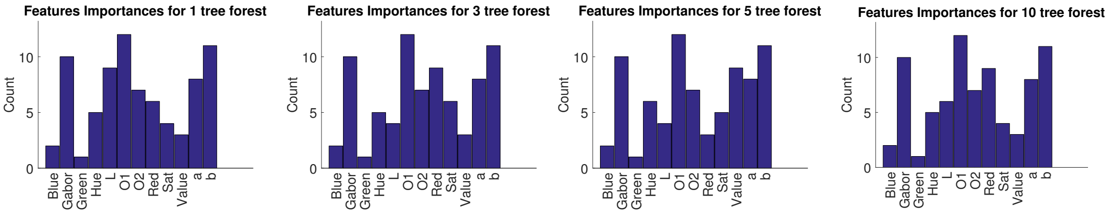
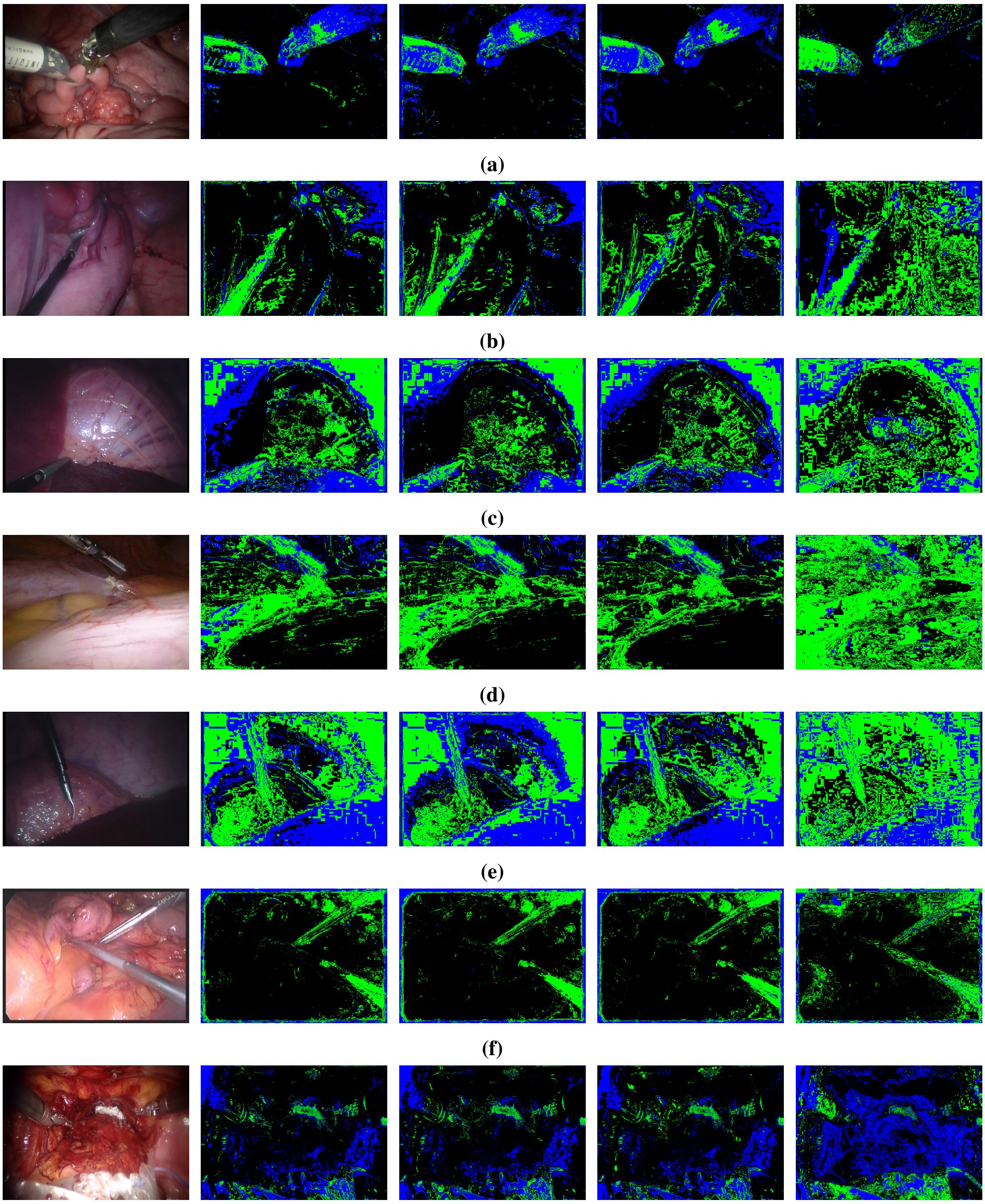


Figure 3.7: Histogram plots of the popularity of different features when training 1, 3, 5, and 10 tree forests using multiple datasets. Features with higher scores remained in the recursive feature analysis procedure for more rounds due to a higher variable importance score. The features O1, Gabor, and Lab channels all feature prominently regardless of the number of trees in the forest.

3.3.2 Single Dataset Evaluation

We are also interested in the performance when classifying a dataset using a RF that has been trained on a minimal subset of data derived from that dataset. This will give important clue as to whether instead of trying to obtain a general classifier which can perform on any dataset it is more effective to train on a minimal training set, where training takes a short enough time to be considered reasonable. In this experiment, we perform a leave-one-out evaluation over each individual dataset, training on the first image of the sequence and then evaluating on the remaining images. As in the multiple dataset evaluation, variable importances for each feature in $\mathcal{X}$ are estimated over all datasets and again we average the score across all of the datasets to obtain a single score for each training round. We demonstrate the accuracy of the forests in Figure 3.9.

As in the multiple dataset evaluation, we again see high precision scores for the background class but also much higher recall, suggesting that the forest is able to distinguish between the ambiguous cases



(a)

(b)

(c)

(d)

(e)

(f)

(g)

Figure 3.8: All rows show an original frame and output from an RF trained on 12, 6, 4 and 2 features respectively from left to right when using a $N-1$ datasets for training and the remaining dataset for validation. Regardless of the number of features the precision of the background remains high, despite the obvious increase in noise, as very few instrument pixels are classified as background. This is seen in the decrease in recall for the background class in Figure 3.6. Datasets with challenging lighting (b-e) show much more noise as the feature number increases as dark pixels cannot be distinguish from the instrument. More well lit sequences (a,f) show less degradation as the number of features is decreased.

more effectively when it is trained on the first image. Precision is still quite low for both instrument classes (≈ 60 % and ≈ 20 %) which is a consequence of large shadow and highlighted regions being mistaken for the instrument class. As with the multiple dataset evaluation, there is minimal advantage to larger feature sets beyond 3 or 4 features with a significant drop off in performance for 2 and 1 feature forests. The example classification images shown in Figure 3.12 and shows a much cleaner segmentation in all datasets.

The feature importances shown in Figure 3.10 are similar to the multiple dataset examples seen in Figure 3.7 where again O1, Gabor and a and b from the CIE Lab color space are the most important features.

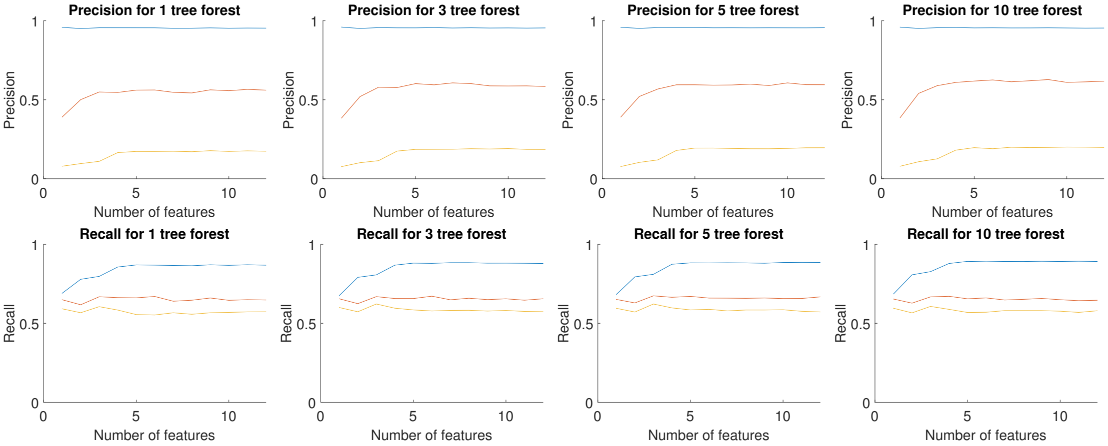

Precision for 1 tree forest
Precision for 3 tree forest
Precision for 5 tree forest
Precision for 10 tree forest
Recall for 1 tree forest
Recall for 3 tree forest
Recall for 5 tree forest
Recall for 10 tree forest

Figure 3.9: Background, Shaft, Head. The precision and recall curves for the 3 target classes when training single dataset RFs with 1, 3, 5 and 10 trees. The values given are the average precision and recall across all 7 datasets when training on the first image from each dataset and evaluating on the remaining images from the dataset. As in the multiple dataset evaluation (see Figure 3.6), the change in precision and recall is limited as the number of trees increases.

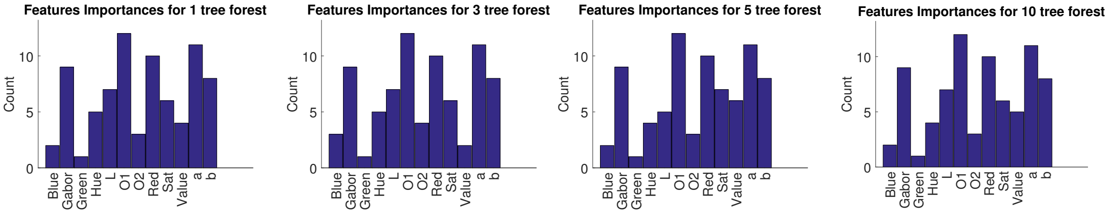
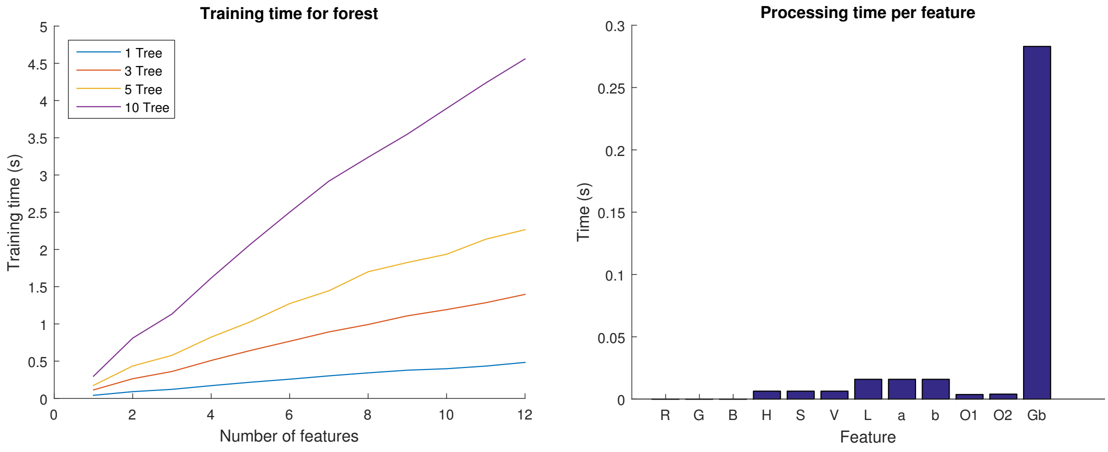

Features Importances for 1 tree forest
Features Importances for 3 tree forest
Features Importances for 5 tree forest
Features Importances for 10 tree forest

Figure 3.10: Histogram plots of the popularity of different features when training 1, 3, 5, and 10 tree RFs using data from the first image of single datasets. As in Figure 3.7, the score for each feature indicates how many rounds of recursive feature elimination it was present in. Features with more importance to the forest remain in the training process for longer. O1, Gabor and CIE Lab a and b are all strong features across the forests with increasing numbers of trees.

We are also interested in the training and evaluation time for single dataset evaluations. In principal training could be performed at the start of a procedure using background pixels generated from the surgical camera when no instruments are in the scene and a set of foreground pixels acquired from the instruments offline. The results in Figure 3.11a show a linear increase in training additional trees and a linear increase in training time as features are added. When studying the evaluation time for each feature, only Gabor features (≈ 0.28 seconds per image) require a noticeable processing time, however they are constructed to respond to edges in the image so the express unique information compared with the other features.

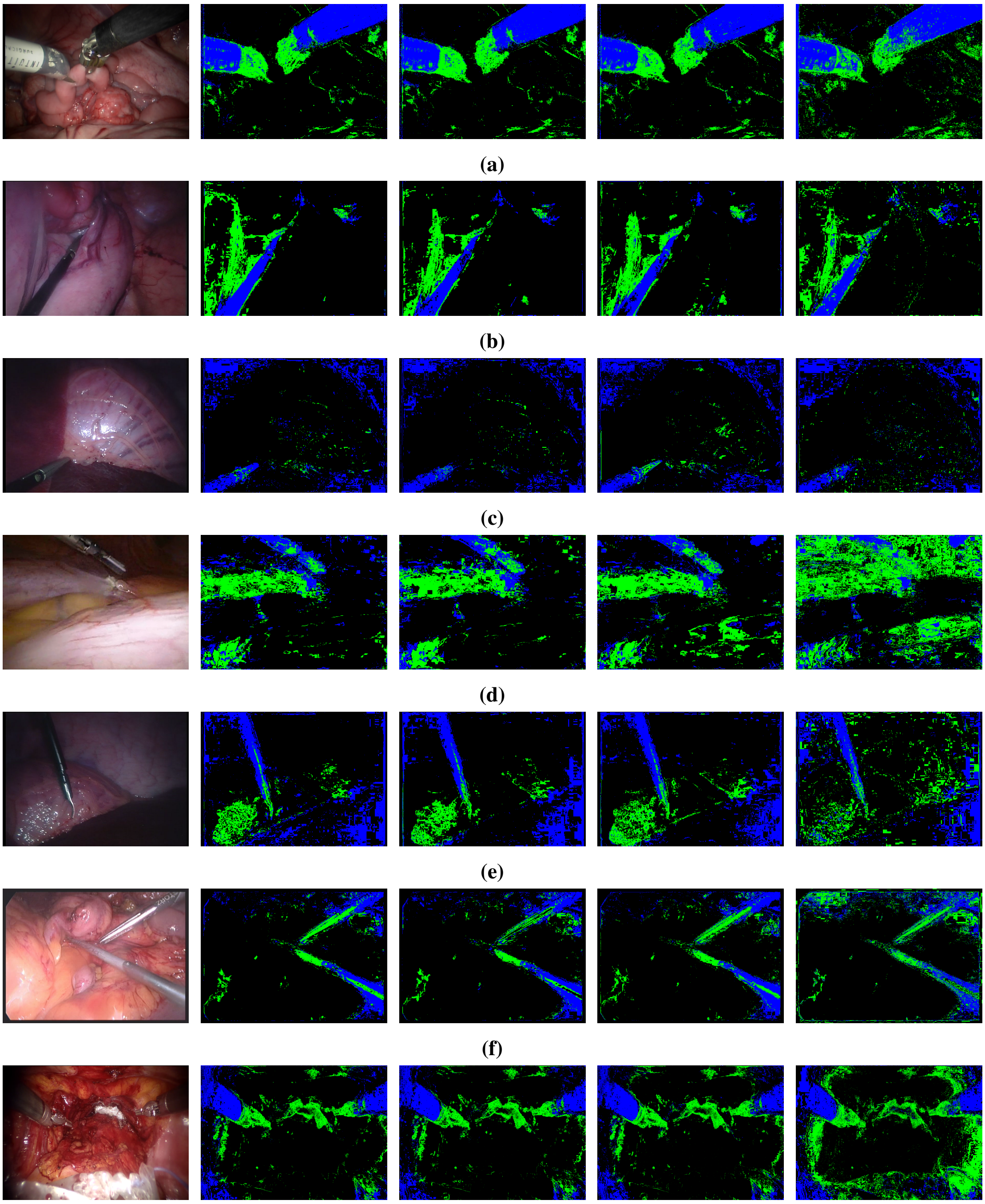


(a) **(b)**

Figure 3.11: (a) The training time for differing numbers of features and different forest sizes when training on single datasets. (b) The time taken to compute each feature for a 720 × 576 image. As the images are converted from YCbCr to RGB on the GPU and all features are computed directly from this color model, we list the time for this feature to be 0. The feature B refers to blue in the RGB model whereas b refers to the chrominance yellow-blue difference channel of the CIE Lab model. Gb refers to the Gabor filter response feature.

3.4 Conclusion

This chapter presents a thorough experimentation with different feature sets to find a feature representation and RF size for surgical instrument segmentation. In the multiple dataset evaluation, precision for the background is observed to be constant at ≈ 0.90 regardless of the number of features however recall begins to drop once 5 or fewer features are used. When this occurs, the forest begins to incorrectly label background pixels as instrument, which is seen particularly in 3.8c, 3.8d and 3.8e. The high precision can be explained by the relatively higher frequency of the background class in the testing set, which means that a metric such as precision which is unaffected by false negatives, is not particularly informative as there are not enough instances of other classes for the false positive count to balance the true positives. The recall measure in this case is much more informative and shows how the performance degrades as the number of features is decreased. As the original data has 3 degrees of freedom and besides the Gabor filter no other feature used in the analysis increases the number of degrees of freedom we would expect limited performance changes after adding further features beyond 4. Most of the information provided by the data can be learned by the forest with a sufficient depth of tree. The precision and recall of the the instrument classes are largely unaffected by the number of features. A decrease in the precision of the shaft is observed at 5 features which corresponds to the drop in recall in the background as pixels in the background begin to be labeled as shaft, increasing the false positive count for the shaft class and increasing the false negative count for the background. Over all classes, the most powerful features were observed to be Opponent 1, b from the CIE Lab color space and the Gabor filter features which were the most popular 3 features when training over all sizes of RF. Green and blue were the least popular 2 features in all sizes of RF and the remaining features had mixed importances in different RF sizes. The overall results demonstrate that simple features such as pixel intensity are not sufficient to achieve good accuracy. The variety of intensities observed in both instruments and tissue pixels requires either neighborhood information to be leveraged or alternatively multiple object classes for different instrument and tissue types to be learned. The dataset size for each multiple dataset evaluation was around 80-90 images with around 10 images of test data. Annotation for this size of dataset can be achieved in less than an hour using image editing software although for much larger datasets, annotation with crowdsourced users has also been demonstrated [137].



(a)

(b)

(c)

(d)

(e)

(f)

(g)

Figure 3.12: All rows show an original frame and output from forests trained on 12, 6, 4 and 2 features respectively from left to right when training on the first image from each dataset and validating on the remaining images. Most datasets show a very limited degradation as the number of features decreases from 12-4 however a large drop in quality is observed when using 2 features. This is particularly noticeable in (d), (e) and (g).

When training a forest to distinguish instruments using a single frame of data we again observed very little change between the precision and recall results when increasing the number of trees in the forest. As with the multiple dataset evaluation, we observe a constant precision score for the background at ≈ 0.95 regardless of the number of features, due to the over presence of background samples in the testing set. Recall for the background and precision for the shaft and head begin to drop off after 3-4 features as portions of the background begin to be mistaken for instrument pixels in the simplified representation. The single dataset evaluated forests show similar feature importances as the multiple dataset evaluation forests with Opponent 1, Gabor and a from CIE Lab all chosen as 3 of the 4 most popular features. The additional feature which proved important is the red channel which was included in the 4 most popular feature in all forest sizes. This channel was not selected highly in multiple dataset training however which suggests that it has high interclass variance between procedures. Despite the minimal performance increase from adding additional trees, we obtain one advantage of extra trees when using random forests, that of being able to obtain probabilistic output from the classification. We can see from a comparison between the precision and recall results of the single dataset evaluation that the performance is increased compared with the multiple dataset evaluation with ≈ 0.5 increase in the shaft precision and ≈ 0.4 and ≈ 0.2 increase in recall for the shaft and head respectively. Combined with the training time of less than 1 second for a single tree forest, our results suggest that for many applications, highly specific forests are a suitable segmentation tool for instrument detection. First frame training and online learning have begun to prove highly popular in computer vision for tracking type tasks as modern processors enable rapid parameter estimation and simplify the recognition task removing complexities from lighting and intraclass appearance variation.

Our conclusion from this analysis is that using single tree or small forests (or potentially other simple models) with minimal feature sets trained on the first frame of data is a valid approach for segmentation of instruments when the end goal is tracking or pose estimation. Training time of less than 1 second is short enough to not be prohibitive and is a justified penalty for the increased performance. For the remainder of this thesis we will focus on single image trained forests of 5 trees using the red, CIE a, Gabor filter and Opponent 1 features.

Chapter 4

Region Based 3D Pose Estimation of Instruments

4.1 Introduction

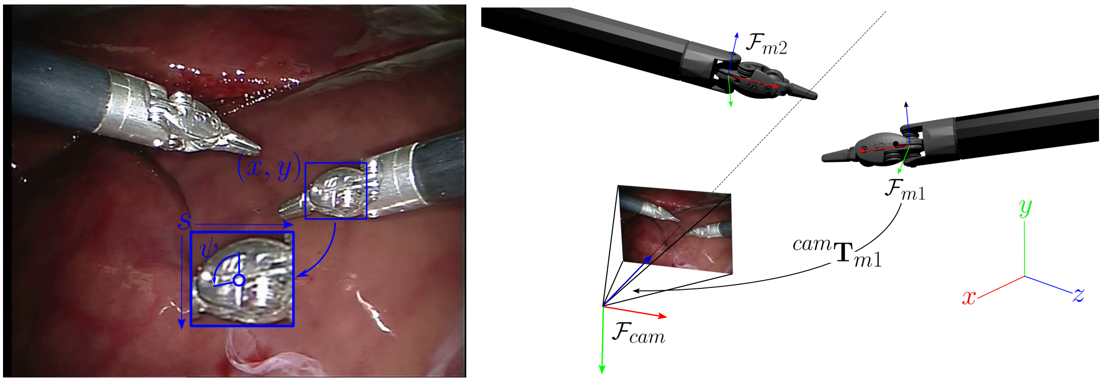
In this chapter we explore the estimation of the 3D pose of surgical instruments directly from 2D images captured by the laparoscopic camera, a problem commonly known as 2D-3D pose estimation. This has important applications for MIS in control, navigation and surgical skills assessment and although many approaches have been proposed to solve the problem, challenges around lighting, motion and the lack of strong instrument features have meant that few methods have achieved the required robustness and accuracy. The majority of techniques covered in the review of the state-of-the-art (see Section 2.3) focussed on pose estimation in 2D, with only a limited number performing full 2D-3D pose estimation [59, 62, 63, 9, 70, 69]. Although computationally and mathematically a more challenging problem, 3D pose estimation holds a number of benefits over its 2D counterpart in allowing more elegant reasoning about how transformations affect appearance, in particular shape when foreshortening occurs, and additionally when occlusions occur between multiple objects. The applications of 3D pose estimation are more extensive as it enables visual servoing and automation with depth correction, which is a critical feature when instruments may move towards sensitive tissue surfaces and blood vessels.

The task of 3D pose estimation involves computing the parameters $\boldsymbol{\theta}$ of a 3D rigid body transform ${}^{cam}\mathbf{T}_{model} = T(\boldsymbol{\theta})$ which maps vertices $\mathbf{X} = [X, Y, Z]^T$ defined in a Euclidean frame of a model coordinate system $\mathcal{F}_{model}$ into the reference frame of the camera $\mathcal{F}_{cam}$. The transform ${}^{cam}\mathbf{T}_{model}$ is composed of a rotation $\mathbf{R} \in \mathbb{SO}3$ and translation $\mathbf{t} \in \mathbb{R}^3$:

$$\mathbf{X}_c = \mathbf{R}\mathbf{X}_m + \mathbf{t} \tag{4.1}$$

where $\boldsymbol{\theta}$ provides the parameters which define both $\mathbf{R}$ and $\mathbf{t}$ and $\mathbf{X}_m$ and $\mathbf{X}_c$ are the coordinates of the same point in $\mathcal{F}_{model}$ and $\mathcal{F}_{cam}$ respectively.

A surgical camera, like any standard consumer camera, cannot record the positions of these vertices directly as 3D points in space, instead making 2D measurements in the projected space of its imaging sensor where the projection is usually modelled as an ideal pinhole camera [113]. This camera model is a linear projection function $\mathbf{K}$ which models how incident light rays from $\mathbf{X}_c$ pass through an aperture at the center of the camera coordinate system $\mathcal{F}_{cam}$ and intersect an imaging plane, represented by the $Z = 1$ plane (see Figure 4.2b). This transform is a perspective projection which transforms between the Euclidean space observed by the camera to the projective space of its imaging system. In projective space, rather than Cartesian coordinates describing the points, an extended system known as homogenous coordinates are used in which all Euclidean points $[X, Y, Z]^T$ have an equivalent represen-



(a) **(b)**

Figure 4.1: (a) An example of the parameters a 2D tracker tries to estimate: (x, y) defines the pixel coordinates of either the center or the corner of a bounding box around the target object, S is a scaling factor (usually relative to the initial bounding box) and ψ is the in-plane rotation angle. (b) An example of a setup that a 3D pose estimation system tries to solve, namely estimating the 3D transform that maps the coordinate system of the target objects onto the camera imaging sensor [8]. $\mathcal{F}_{mx}$ refers to the frame of the instrument where x is a numerical index to distinguish different instruments. The same naming is used for the transform to this model $^{cam}\mathbf{T}_{m1}$. The image plane shown in the image represents the $Z = 1$ plane in the camera frame $\mathcal{F}_{cam}$.

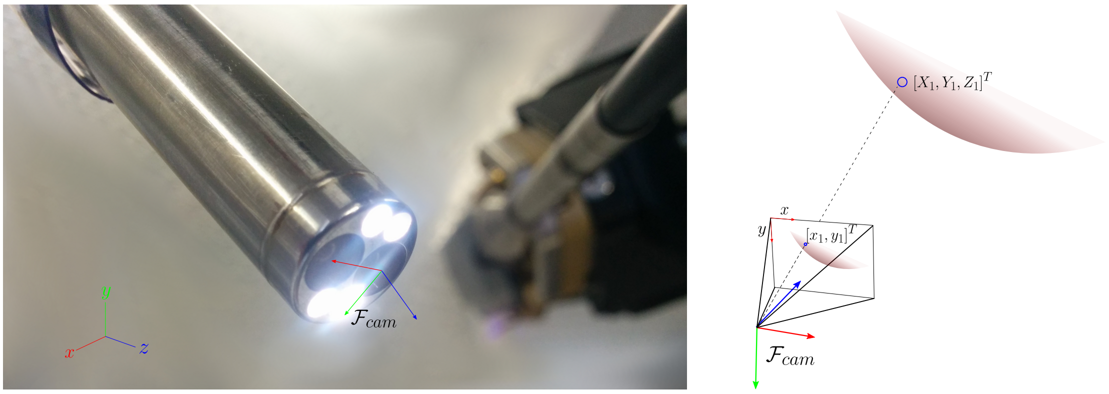
tation $[X/Z, Y/Z, 1]^T$ and all points in the 2D Euclidean plane $[x, y]^T$ have an equivalent representation $[xZ, yZ, Z]^T$. The camera projection function K can by written as:

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \mathbf{K} \begin{bmatrix} X/Z \\ Y/Z \\ 1 \end{bmatrix} = \mathbf{K} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{4.2}$$

where the equivalence between the point $[X, Y, Z]^T$ and $[X/Z, Y/Z, 1]^T$ allowed in homogenous coordinates is used to represent the 3D points projected on the $Z = 1$ plane where the imaging sensor is defined and $[x, y, 1]^T$ describes the coordinates of the projected points in terms of the pixels of the imaging sensor. The camera projection model $\mathbf{K}$ [113] is defined as:

$$\mathbf{K} = \begin{bmatrix} f_x & \gamma & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{4.3}$$

where (f_x, f_y) represents the camera focal length f in units of pixels and as non-square pixels are common on most camera sensors, a different value for f in the x and y dimension occurs. (c_x, c_y) is the camera principal point which measures the pixel location at which the optical axis of the imaging system intersects the image sensor allowing for cases when the center of the imaging coordinate system does not align exactly with the center of the camera coordinate system. γ allows for skew in image plane where the x and y axis are not perpendicular and is normally zero. This linear projection function is normally augmented with a non-linear distortion model to account for warping to the image caused by the curvature of the camera lens. The most common approach is to use a polynomial model [138] to describe this effect and by using this model, a real camera image is typically unwarped so that it appears as if it were captured by an ideal perspective camera [129]. This modelling allows the formulation of the problem we are trying to solve in 2D-3D pose estimation to be described mathematically as a linear inverse problem where we have a series of N measurements, one for each pixel on the camera imaging sensor, where some $n \subset N$ represent projections of the vertices of our model and the remaining $n' \setminus n$



(a) **(b)**

Figure 4.2: (a) The camera coordinate system of a stereo laparoscope used in MIS. The camera looks down the z axis of the right-handed coordinate system with y pointing down. (b) The pinhole projection model.

pixels contain light incident from other objects in the scene and the background. This inverse problem can be written as:

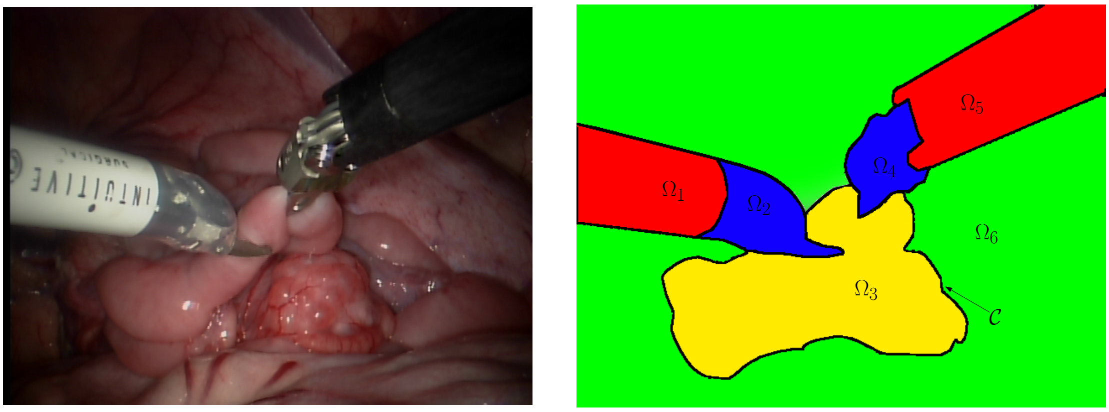
$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \mathbf{K}\ {}^{cam}\mathbf{T}_{model} \begin{bmatrix} X_m \\ Y_m \\ Z_m \end{bmatrix} \tag{4.4}$$

with unknowns $\mathbf{K}$ and ${}^{cam}\mathbf{T}_{model}$. The parameters of $\mathbf{K}$ can be determined using a process known as intrinsic camera calibration [138] so the challenge in pose estimation reduces to the estimation of $\mathbf{T}$ and the finding of correspondences between the model points $\mathbf{X}_m$ and the measurements on the image sensor. Arguably the most challenging component of this process is finding correspondences as only an unknown subset m of the model vertices are visible from a single camera view and additionally the subset n of pixels that m projects to is unknown. Assuming these two subsets are known, finding which elements of n match to each element of m is again hugely complex. These correspondences can either be constructed as localized points or using collections of pixels such as interior or exterior gradients or the shape of distinct regions [139]. The point based methods use an explicit one-to-one matching between pixel locations and model points where each correspondence is extremely informative in estimating ${}^{cam}\mathbf{T}_{model}$ but correspondences are routinely hard to find and match accurately. Edge and region based methods soften the correspondence problem by attempting to match ensembles of pixels to ensembles of model points. In each case, a one-to-one match is not explicitly sought, rather a membership type matching is applied when a transform that causes each member of the pixel ensemble to match to a single point in the model ensemble. In these cases, the matching criteria is much simpler and can accommodate larger inaccuracies but the challenge in this case is that each match provides much less informative power in estimating ${}^{cam}\mathbf{T}_{model}$.

In the following sections we explore how the 3D pose of a rigid surgical instrument can be estimated from a single image. To achieve this we draw on successes in region based methods of 3D pose estimation which have been demonstrated in the medical imaging literature as well as the wider computer vision community to be robust to noise and motion blur which have impacted more sensitive edge and point based methods [140]. To simplify the problem we do not consider articulation of robotic instruments or the clasper opening of laparoscopic instrument, instead treating the instrument as a rigid body. However, we build a method which retains the potential to incorporate articulated motion without major modification and additionally allows easy extension to handle instruments that are holding additional imaging devices, such as pick up ultrasound probes. Our contributions in this chapter are the extension of single region type 3D tracking methods to account for multiple homogenous regions on surgical instruments and we demonstrate that this provides improvements in the pose estimation in the cases of occlusion and noise. We achieve this by providing extensive experiments on calibrated ex-vivo data and qualitative in-vivo data. The work presented in this chapter was described in the publications [126, 141, 142].

4.2 3D Region Based Pose Estimation with Level Sets

We begin by looking at how 3D pose can be estimated by partitioning an image into regions. Region segmentation extends the pixel based segmentation techniques developed in Chapter 3 as it assumes a consistency in labeling within distinct areas of image rather than independent labels for each pixel of the image. It achieves this goal by taking a top-down view of the problem whereby the individual pixel appearance that drives the pixel based segmentation is combined with priors to force the regions to conform to an expected shape and this enables ambiguities in the pixel information to be resolved. These priors can be applied using learned 2D shape spaces [143] although a useful alternative is to directly generate them by concurrently solving 3D pose estimation using the pixel appearances and using the shape of the projected model as a region division. In the remainder of this section, we look at how images can be segmented up into regions and how misalignments between these regions and color models can be reduced by moving the boundaries in an efficient manner. We then end the section by looking at how these boundary movements can be linked directly to estimation of 3D pose parameters.



(a) **(b)**

Figure 4.3: An example frame Ω is divided up into regions $\Omega_1, ..., \Omega_6$ and contour $\mathcal{C}$. Each region represents a semantically distinct area of the image, where Ω_1 and Ω_5 represent instrument shafts, Ω_2 and Ω_4 represent instrument claspers and Ω_3 and Ω_6 represent tissue samples.

4.2.1 Region Based Image Segmentation

Region based segmentation techniques aim to compute the decomposition of an image domain Ω into regions Ω_i and a boundary $\mathcal{C}$ as:

$$\Omega = \Omega_1 \cup \Omega_2 \cup ... \cup \Omega_n \cup \mathcal{C} \tag{4.5}$$

such that a similarity term is respected within each Ω_i and is discontinuous across the boundary [144]. Unsupervised similarity can be used to find local clusterings in the feature space of the image [145, 146] or alternatively it can be imposed by statistical models which can either be learned online [85] or offline [147]. Statistical model have become increasingly popular as part of tracking frameworks as they allow greater robustness to drift and occlusion compared with unsupervised methods.

There are two broad approaches for solving this decomposition problem, based either on spatially discrete approaches [148, 149] which describe the image pixels as nodes in a undirected graph and attempts to find optimal cuts in this graph [150] or spatially continuous approaches which seek to deform boundaries and contours in the image using variational methods [151]. As with all variational methods,

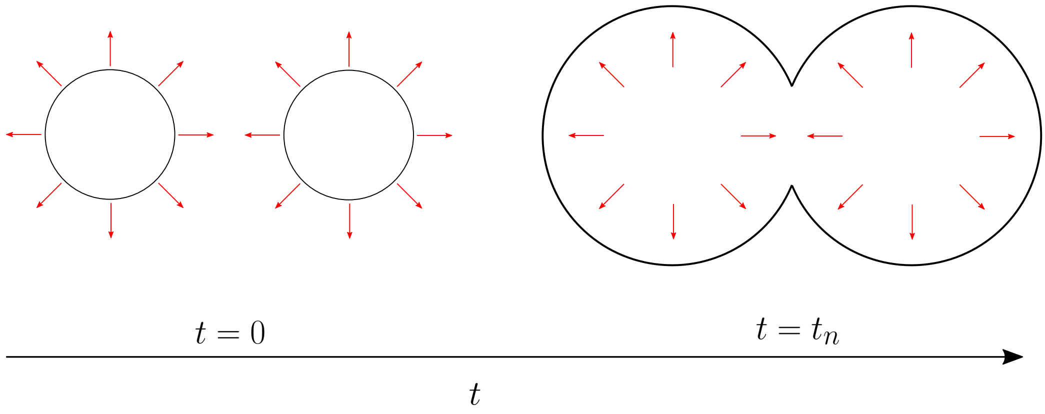


Figure 4.4: The front propagation from $t = 0$ to $t = t_n$ when the 2 distinct fronts propagate and join together. The regions are shown by the blue pixels and the contour $\mathcal{C}$ evolves so that it correctly divides the regions from the surrounding white pixels.

this involve optimizing an energy functional which for image segmentation is normally defined as [152]:

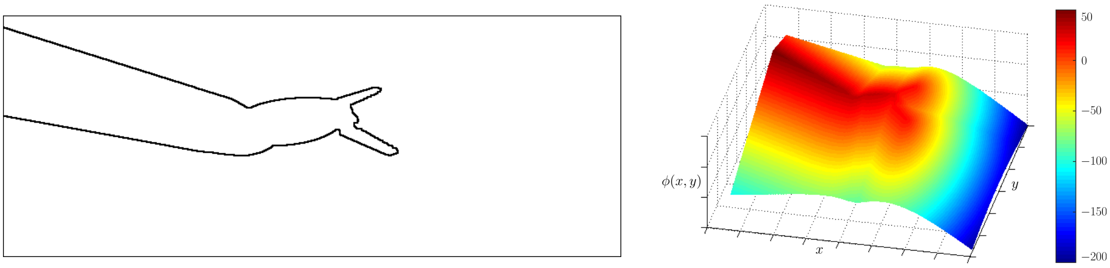
$$E(\mathcal{C}) = \sum_{\Omega_i \in \Omega} \int_{\Omega_i} f(I(\mathbf{x}), \chi_i) \, \mathrm{d}\mathbf{x} \tag{4.6}$$

where $f(I(\mathbf{x}), \chi_i)$ is the statistical model for the i^{th} region defined by a partitioning of the contour $\mathcal{C}$. Each statistical model is dependent on appearance parameters for the i^{th} region χ_i and the image features at $I(\mathbf{x})$. This region based representation of segmentation differs from the well known edge based segmentations [153] which normally perform the integral along the contour where a function $f(.)$ responds to intensity changes in the image. Optimization of the energy E occurs when the image data in the regions defined by $\mathcal{C}$ agrees maximally with the statistical model for that region and is achieved through the evolution of the $\mathcal{C}$ (see Figure 4.4) with a measure of velocity or *flow* along the normal vector to each point on the contour [154]. A central component to any method for solving the optimization is the representation of $\mathcal{C}$ and the earliest methods used parametric curves which described a Lagrangian formulation of the flow. This involves modelling the curve explicitly in terms of several control points upon which the optimization routine acts and the most well known of these are geometric active contours, commonly known as snakes [153]. The main limitation of the Lagrangian formulation is that it imposes a constraint of a static topology on the boundary and numerically these have been shown to be quite unstable when the particles begin to overlap one another or sharp corners appear [155]. As a solution to these problems, an alternative Eulerian representation of the flow has been presented [156] which describes its value at fixed points on a grid over the image. This was achieved by embedding $\mathcal{C}$ in a one dimensional higher surface ϕ effectively describing the shape of the evolving contour implicitly, at different constant levels of ϕ, known as level sets. Level sets of function ϕ define the set of points:

$$\mathcal{C} = \{x \in \Omega | \phi(x) = C\} \tag{4.7}$$

where C is a constant value, often set to 0 in which case $\mathcal{C}$ is known as a zero level set. This effectively changes the problem from attempting to track a propagating contour to instead tracking ϕ.

There are many embedding functions ϕ that solve the equations of motion for the curve, the requirements being that the function be smooth and Lipschitz continuous as well as positive inside the contour,



(a)

(b)

Figure 4.5: A contour (a) is used to generate a SDF ϕ where each pixel takes on the Euclidean distance to the nearest contour point with a negative sign applied outside the $\mathcal{C}$. This is shown projected into 3D (b) and colormapped for clarity. A SDF from the contour of a robotic surgical instrument. Each value in ϕ is the distance from the (x, y) coordinate to the nearest contour point.

zero at the contour and negative outside the contour:

$$\phi(\mathbf{x}) = \begin{cases} > 0 & \text{if } \mathbf{x} \text{ inside } \mathcal{C} \\ < 0 & \text{if } \mathbf{x} \text{ outside } \mathcal{C} \\ = 0 & \text{if } \mathbf{x} \in \mathcal{C} \end{cases} \tag{4.8}$$

The most popular choice which fulfils this criteria is the signed distance function (SDF) which is defined as:

$$\phi(\mathbf{x}) = \begin{cases} d(\mathbf{x}, \mathcal{C}) & \text{if } \mathbf{x} \text{ inside } \mathcal{C} \\ -d(\mathbf{x}, \mathcal{C}) & \text{if } \mathbf{x} \text{ outside } \mathcal{C} \\ 0 & \text{if } \mathbf{x} \in \mathcal{C} \end{cases} \tag{4.9}$$

where $d(\mathbf{x}, \mathcal{C})$ returns the Euclidean distance from $\mathbf{x}$ to the closest point on $\mathcal{C}$. The SDF obeys the additional criteria:

$$|\nabla \phi| = 1 \tag{4.10}$$

In Equation 4.6 the energy is defined in terms of the contour where sums are performed over different regions proposed by $\mathcal{C}$. When using level sets to represent the contour, region interiors are represented by positive values and exteriors by negative values. This enables membership of the regions to be expressed with a Heaviside function [152] and in the simplified case of a 2 class segmentation, this allows the energy to be written as:

$$E(\phi) = \int_{\Omega} H(\phi(\mathbf{x})) f(I(\mathbf{x}), \chi_0) + (1 - H(\phi(\mathbf{x}))) f(I(\mathbf{x}), \chi_1) \, \mathrm{d}\mathbf{x} \tag{4.11}$$

where $H(.)$ is the Heaviside function which transforms the values of ϕ into membership values as:

$$H(\phi) = \begin{cases} 1 & \text{if } \phi \geq 0 \\ 0 & \text{else} \end{cases} \tag{4.12}$$

This equates to a computationally efficient system where a sum over the pixels in Ω scores them with the appropriate pixel similarity function due to the vanishing opposing term. A common modification to the Heaviside function is to use a smoothed approximation, which allows for a degree of uncertainty in estimating the location of the contour [109]. An efficient approximation is computed as [157]:

$$H(\phi) = \begin{cases} \frac{1}{2}\left(1 + \frac{\phi}{\alpha} + \frac{1}{\pi}\sin(\frac{\pi\phi}{\alpha})\right) & \text{if } |\phi| \leq \alpha \\ 1 & \text{if } \phi > \alpha \\ 0 & \text{if } \phi < -\alpha \end{cases} \tag{4.13}$$

where $\alpha = 3$ is a suitable choice for the boundary width. After each contour evolution, the function ϕ no longer represents a SDF. There are numerous methods of reinitializing ϕ such as a brute force re-computation and the fast marching methods.

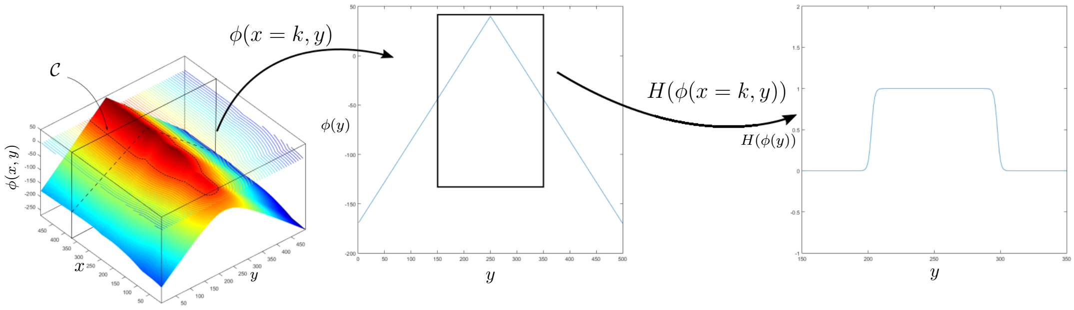


Figure 4.6: The SDF of the closed contour $\mathcal{C}$ is computed as $d(\mathbf{x}, \mathcal{C})$ where $d(.)$ returns the Euclidean distance. This is then transformed into region membership terms with a Heaviside function, or an approximation of this.

4.2.2 3D Pose Estimation as Region Based Level Set Segmentation

Building upon these powerful variational segmentation frameworks has become popular as a method of estimating the 3D pose of objects [158, 110, 147, 140, 159]. By linking the shape of the segmented regions directly to the poses that could generate them, it becomes possible to describe the pose estimation process fully within a segmentation framework. Many methods [159, 160] perform the step of estimating the pose independently from the segmentation by effectively forming direct correspondences between the projected silhouette of the 3D model and $\mathcal{C}$. However, [161, 158] proposed an elegant method of enforcing a strict constraint on the segmentation shape by constructing the contour (or ϕ) directly from a projection of the 3D model by parametrizing it with the rigid body pose parameters $\boldsymbol{\theta}$. This formulation is greatly simplified over working with an infinite dimensional contour as it does not require complex regularizations to maintain a suitable shape. Imposing this type of shape constraint can be seen as a strict shape prior, enabling regions of the image where the object is occluded to be successfully segmented. Bayesian approaches using learned shape spaces have also been used to this end [109, 162, 143].

4.2.3 Multi-Region Level Sets for Robotic Surgical Instruments

Our RF work in Chapter 3 provides us with a robust method of assessing the similarity function $f(I(\mathbf{x}), \chi_i)$ and the fact that we classify the instrument as multiple regions allows us to consider two types of region based pose estimation techniques. The first is the classic binary foreground/background model that is most common in 3D pose estimation with level sets which allows the problem to be cast as contour matching using silhouettes. This simplification affords a great deal of invariance with respect to the chosen object and typically works well when the appearance model between foreground and background is strong, resulting in a clean contour. However, for manufactured robotic instruments, this simplification ignores strong internal homogeneous regions our RF detects which can be useful in generating an additional strong delineating contour. A particular advantage of this additional contour is that it constructs a fully visible single contour, which is not the case for a binary silhouette that intersects

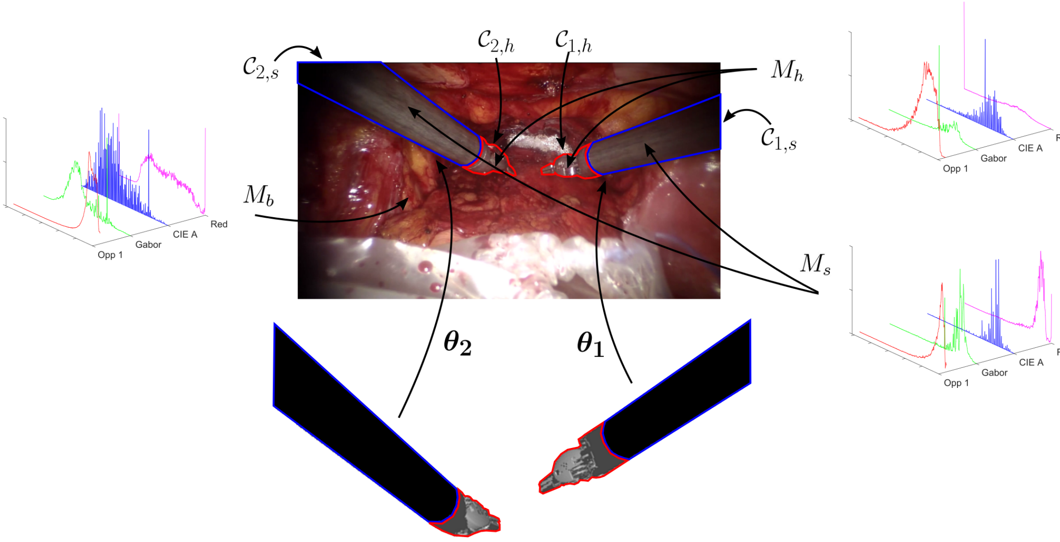


Figure 4.7: The color models $M_{s,h,b}$ describe the multiple interior and single exterior regions of the contours $\mathcal{C}_{1,2}$. These contours are generated from sampling in the pose space of the two instruments $\boldsymbol{\theta}_{1,2}$. Only poses that are consistent with projections of the 3D models are allowed, leading to much greater efficiency than classical solutions which allow the contour to evolve with a time parameter.

the edge of the image, and this can in principal provide information about foreshortening. For example, when the instrument rotates around the vertical axis of the image, the appearance change in the silhouette is limited compared with a case when the whole instrument is fully visible where there would be a noticeable reduction in the size of the instrument. With a partially visible contour, a translation along the axis of the instrument towards the intersection point of the image would be a plausible explanation for the observed appearance change. However, if a fully visible contour around the instrument's metallic end is present then this translation would cause a misalignment between the region boundary between the shaft and the metal clasper. A further advantage of tracking the instrument head as a separate region arises when common occlusions of robotic instruments occur, such as when tissue moves in front of the instrument which normally happens at the tip as this is the part designed for tissue interaction. A single-region level set would be pushed backwards along the central axis of the shaft to explain this occlusion as this particular pose update would best explain the observation. However, when using a multiple-region level set, the intensity change between the shaft and the head can in principal prevent this motion as it provides a constraint along the axis of the instrument. Using multiple regions for each instrument involves extending the cost defined in Eq. 4.6 as:

$$E_{region}(\boldsymbol{\theta}) = -\sum_{i}^{K} \int_{\Omega} H(\phi^k(\mathbf{x},\boldsymbol{\theta})) f(I(\mathbf{x}),\chi_k) + (1 - H(\phi^k(\mathbf{x},\boldsymbol{\theta}))) f(I(\mathbf{x}),\chi_{n(k)}) \,\mathrm{d}\mathbf{x} \tag{4.14}$$

where the terms in the equation are the same except we now require an additional sum over the K regions where each region is defined by its own color model $f(I(\mathbf{x}),\chi_k)$. Rather than performing one-against-all for the background distribution, we instead use the expected neighbour class $n(k)$ of the pixel $\mathbf{x}$ as the chosen background distribution $f(I(\mathbf{x}),\chi_{n(k)})$. If a stereo camera is used in the procedure, we can trivially add stereo constraints to this cost function by projecting the model into both camera images using a pre-computed extrinsic camera calibration [110]. The set of model parameters remains the same for both views in this case. When tracking multiple instruments, we can mask out regions of the image that we expect to be occluded by another tracked instrument. We achieve this by maintaining a depth buffer of all tracked instruments in the scene and only adding a contribution from a pixel if the model point that is expected to have generated it is in view.

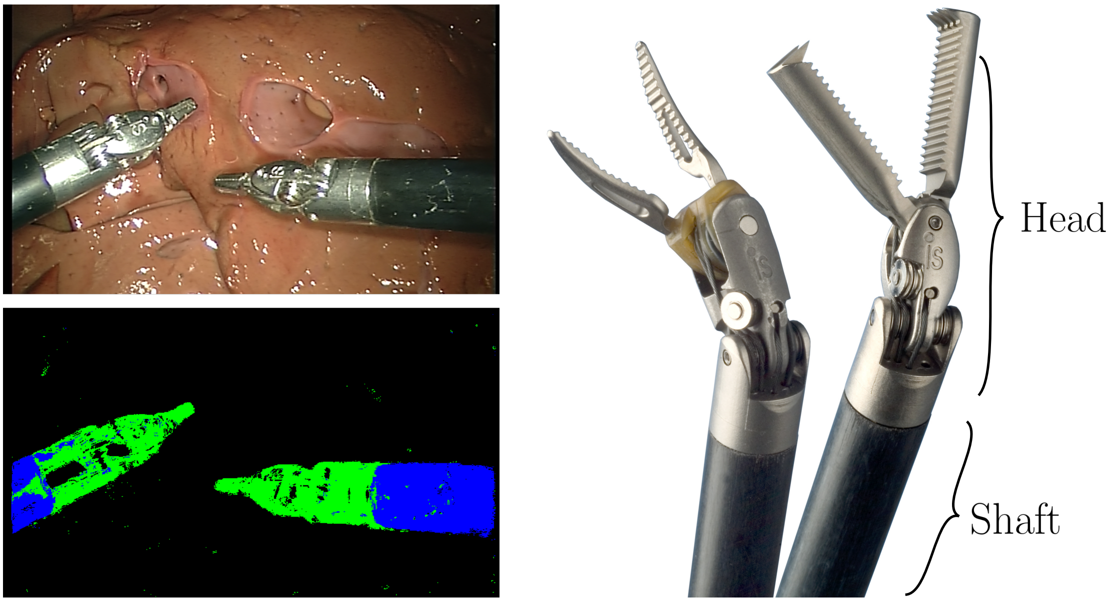


Figure 4.8: (a) The feature distribution for each of the $K = 3$ classes with output classification. (b) The typical shaft/head divide for many robotic surgical instruments.

4.3 Optimization

There are many forms of optimization strategy available. We use gradient descent to optimize E, which is the most popular choice in 3D object tracking [110, 143, 163, 140, 159] for optimization owing to its simple requirement of first derivatives. For the region based energy, this is computed as:

$$\frac{\partial E(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \sum_{k \in K} \sum_{\mathbf{x} \in \Omega} \frac{\partial}{\partial \boldsymbol{\theta}} - \log\left(H(\phi^k(\mathbf{x}, \boldsymbol{\theta}))f(I(\mathbf{x}), \chi_k) + (1 - H(\phi^k(\mathbf{x}, \boldsymbol{\theta})))f(I(\mathbf{x}), \chi_{n(k)})\right) \tag{4.15}$$

$$= \sum_{k \in K} \sum_{\mathbf{x} \in \Omega} \frac{f(I(\mathbf{x}), \chi_k) - f(I(\mathbf{x}), \chi_{n(k)})}{H(\phi^k(\mathbf{x}, \boldsymbol{\theta}))f(I(\mathbf{x}), \chi_k) + (1 - H(\phi^k(\mathbf{x}, \boldsymbol{\theta})))f(I(\mathbf{x}), \chi_{n(k)})} \frac{\partial H}{\partial \boldsymbol{\theta}} \tag{4.16}$$

where

$$\frac{\partial H}{\partial \boldsymbol{\theta}} = \delta(\mathbf{x}) \left[\frac{\partial \phi^k(\mathbf{x}, \boldsymbol{\theta})}{\partial x} \frac{\partial x}{\partial \boldsymbol{\theta}}, \frac{\partial \phi^k(\mathbf{x}, \boldsymbol{\theta})}{\partial y} \frac{\partial y}{\partial \boldsymbol{\theta}} \right] \tag{4.17}$$

where $\partial \phi^k(\mathbf{x}, \boldsymbol{\theta}/\partial x, y)$ can be computed using finite differences and $\delta(.)$ is the derivative of the smoothed Heaviside function and corresponds to a smoothed Dirac delta function which has the effect of weighting the derivative terms so that only the points around the contour contribute to the optimization. This term can be approximated as [157]:

$$\delta(\mathbf{x}) = \begin{cases} \frac{1}{2\alpha}\left(1 + \cos\frac{\pi \mathbf{x}}{\alpha}\right) & \text{if } |\mathbf{x}| \leq \alpha \\ 0 & \text{if } |\mathbf{x}| > \alpha \end{cases} \tag{4.18}$$

Equation 4.16 requires derivatives of 3D model to 2D camera point projections $[X, Y, Z]^T \mapsto [x, y]^T$ with respect to the pose parameters of the transform ${}^{cam}\mathbf{T}_{model}$ which evaluate to:

$$\frac{\partial x}{\partial \theta_i} = f_u \frac{1}{Z^2}(Z\frac{\partial X}{\partial \boldsymbol{\theta}_i} - X\frac{\partial Z}{\partial \boldsymbol{\theta}_i}) \tag{4.19}$$

$$\frac{\partial y}{\partial \theta_i} = f_v \frac{1}{Z^2}(Z\frac{\partial Y}{\partial \boldsymbol{\theta}_i} - Y\frac{\partial Z}{\partial \boldsymbol{\theta}_i}) \tag{4.20}$$

In our work, we represent the rotation $\mathbf{R}$ as a quaternion [164], which despite being an over-parametrization of the rotation, has a number of useful advantages over comparable Euler angle and angle-axis representations such that normalization is achieved with a simple Euclidean norm and an inverse is computed as by negation of just 3 components [165]. The quaternion $\mathbf{q}$ can represent a rotation matrix R as:

$$R(\mathbf{q}) = \begin{bmatrix} 1 - 2q_y^2 & 2(q_xq_y + q_wq_z) & 2(q_xq_z - q_wq_y) \\ 2(q_xq_y - q_wq_z) & 1 - 2q_x^2 - 2q_z^2 & 2q_yq_z + 2q_wq_x \\ 2(q_xq_z + q_wq_y) & 2(q_yq_z - 2q_wq_x) & 1 - 2q_x^2 - 2q_y^2 \end{bmatrix} \tag{4.21}$$

where q_w forms the scalar part of q and $[q_x, q_y, q_z]$ form the vector component [166]. This results in a Jacobian for each 3D model point in camera coordinates $\mathbf{X}_c = [X_c, Y_c, Z_c]^T$ for the translation DOFs $[t_x, y_y, t_z]^T$ as:

$$\begin{bmatrix} \frac{\partial X_c}{\partial \boldsymbol{\theta}} \\ \frac{\partial Y_c}{\partial \boldsymbol{\theta}} \\ \frac{\partial Z_c}{\partial \boldsymbol{\theta}} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.22}$$

and for the rotation parameters $[q_w, q_x, q_y, q_z]$ as:

$$\begin{bmatrix} \frac{\partial X_c}{\partial \boldsymbol{\theta}} \\ \frac{\partial Y_c}{\partial \boldsymbol{\theta}} \\ \frac{\partial Z_c}{\partial \boldsymbol{\theta}} \end{bmatrix} = \begin{bmatrix} 2q_yZ_m - 2q_zY_m & 2q_wY_m + 2q_zZ_m & 2q_xY_m - 4q_yX_m + 2q_wZ_m & 2q_xZ_m - 2q_wY_m - 4q_zX_m \\ 2q_zZ_m - 2q_xZ_m & 2q_yX_m - 4q_xY_m - 2q_wZ_m & 2q_0X_m + 2q_zX_m & 2q_wX_m - 4q_xY_m + 2q_yZ_m \\ 2q_xY_m - 2q_yX_m & 2q_zX_m + 2q_wY_m - 4q_xZm & 2q_zY_m - 2q_wX_m - 4q_yZ_m & 2q_xX_m + 2q_yY_m \end{bmatrix} \tag{4.23}$$

4.4 Scaling Between Rotation and Translation

Performing gradient based searches over the space of rigid body transforms is challenging because the special Euclidean group $SE(3)$ which represents the rigid body transforms is non-metric which means there exists no scaling of the dimensions of the space that allows a valid metric to be defined [165]. This occurs because the gradient of a vertex position with respect to rotation is affected by how far from the center of rotation the vertex lies, meaning that vertices further from the center of rotation will have a much larger effect on the total Jacobian. To solve this scaling imbalance, the center of the coordinate system can be positioned near to the instrument tip which means that the vertices that contribute to the Jacobian are approximately equidistant to the center of rotation and have similar magnitudes to one another. Another challenge that occurs when working with gradients of rotations and translations is dealing with the difference in units between them. Methods for dealing with this can involve scaling the coordinate system to a unit square [165] but a more straightforward way to scale the rotations and translation when dealing with objects of known size is to pre-scale the rotations and translations to a step size manually. To account for desiring smaller steps as we approach the minimum, we choose 0.008 radians for the rotation step and translation and 0.18mm for the first $N/2$ steps and 0.002 radians and 0.04mm for the last $N/2$ steps where N is the maximum number of gradient descent steps, which is normally set to 15 but can be configured at runtime.

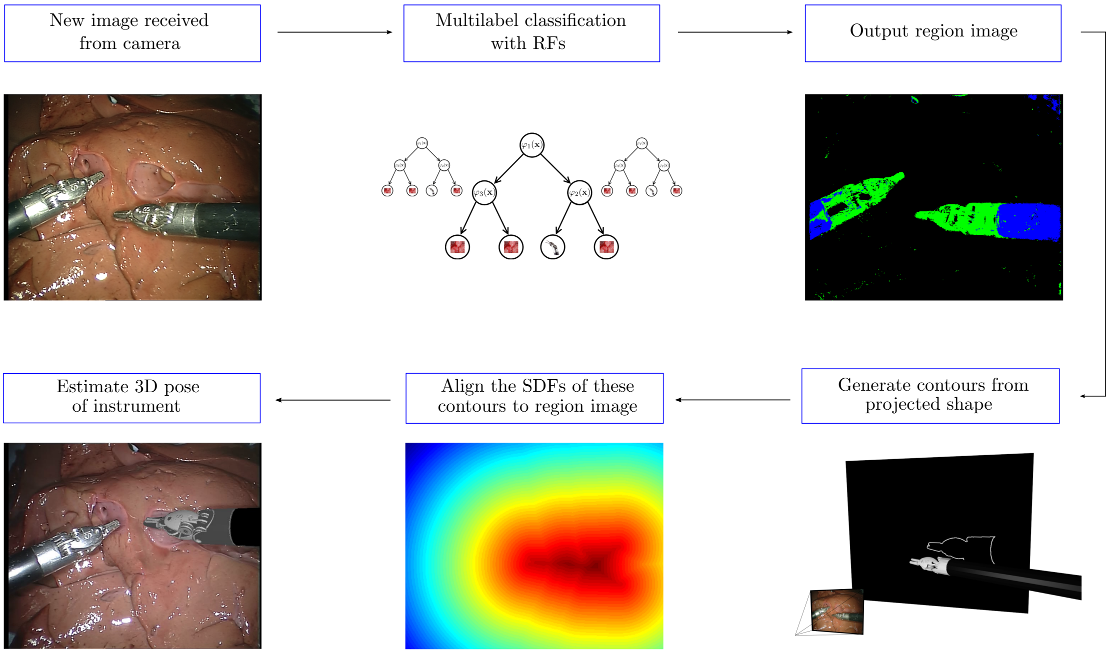


Figure 4.9: An overview of our method. Following the arrows around the flow chart, the images captured by the surgical camera are classified with a multiclass RF and using this output region image, the 3D pose is estimated by generating a contour and subsequent SDF and aligning this to the classifier output.

4.5 Temporal Tracking

Frame-to-frame tracking is provided with a linear Kalman filter for both position and orientation. In the Kalman filter, orientation is transformed from the quaternion representation to the extrinsic Euler angle representation $[r_x, r_y, r_z]$, where each term refers to a rotation around the x, y and z axis of the camera coordinate system respectively. The advantage of performing this step over using the quaternion representation directly is that the Euler representation is linear allowing the linear Kalman filter to be used. Our pose estimation for the k^{th} estimate is therefore defined as:

$$\boldsymbol{\theta}_k = [t_x, t_y, t_z, \dot{t_x}, \dot{t_y}, \dot{t_z}, r_x, r_y, r_z, \dot{r_x}, \dot{r_y}, \dot{r_z}] \tag{4.24}$$

where the terms have their usual meanings. We update pose using the standard Kalman Filter equations:

$$\boldsymbol{\theta}_k = \mathbf{F}\boldsymbol{\theta}_{k-1} + N(0, \mathbf{Q}_s) \tag{4.25}$$

$$\boldsymbol{\theta}'_k = \mathbf{M}\boldsymbol{\theta}_k + N(0, \mathbf{Q}_o) \tag{4.26}$$

where $\boldsymbol{\theta}'_k$ is the measurement vector, $\mathbf{F}$ is the position-velocity state transition matrix and $\mathbf{M}$ is the identity observation model. Both are corrupted by normally distributed noise of zero mean and covariance $\mathbf{Q}_s$ for the state and $\mathbf{Q}_o$ for the observation.

4.6 Experiments

4.6.1 Implementation Details

For convenience, homogenous regions are manually selected rather than learned with a clustering algorithm. We create a texture map for a 3D mesh of our target model, which is a da Vinci LND instrument, where each texel contains an integer label for a particular homogenous region. We use OpenGL and OpenGL shader language (GLSL) to render the contour, predicted label map and 3D vertex coordinates needed for the optimization and we use a CUDA implementation of the signed distance function computation [167]. The remainder of our algorithm is written in C++ and makes use of the OpenCV library [129] and Cinder[1] which provides a lightweight cross-platform wrapper for OpenGL. Processing time measured on a single core of a 1.9GHz processor for classification of a single stereo frame using a RF is ≈ 0.83 seconds, for a gradient descent step on one stereo frame is ≈ 0.3 seconds (typically 10-20 steps required). The most computationally expensive component of the method is the sum-over-pixels in the region based Jacobian for which each pixel contributes to the sum independently allowing for real time speeds when using a GPU implementation [110]. Furthermore, RFs are suitable for GPU parallelization and by only performing classification in regions where the derivatives are non-zero, we can greatly reduce the number of pixels which require classification to around 0.5 $\%$ of the image.

4.6.2 Dataset Construction

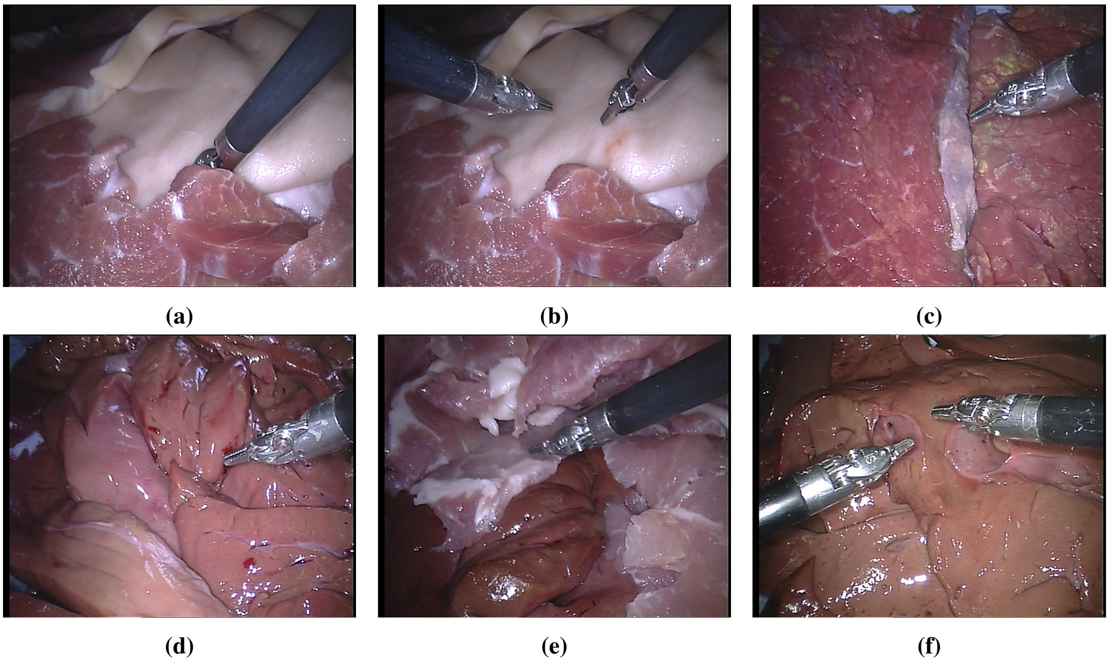


Figure 4.10: Example frames from 6 of the 7 datasets used in our evaluation. Each dataset was captured using a da Vinci classic 720×576 stereo laparoscope at 25Hz with kinematics provided by the the DVRK control system.

Obtaining ground truth for 3D pose estimation is highly complex with marker based tracking systems, such as optical trackers and electromagnetic trackers requiring complex calibrations between the camera and model based coordinate systems and the coordinate systems of the markers. One alternative when tracking robotic instruments is to read the joint encoder status and use forward kinematics to compute the relative pose between the camera and instrument coordinate system. However, as the cable driven kinematics of systems such as da Vinci contain significant absolute position errors, these errors must be eliminated to obtain an estimate of the ground truth pose of the instrument. To this end, we designed a capture system which enables us to collect video data from the da Vinci camera and synchronously capture joint encoder data using the da Vinci Research Kit[2] (DVRK) [168] and Intuitive Surgical Inc. API [169]. This system renders instrument models using forward kinematics in the reference frame of the calibrated stereo camera and by manually correcting the joint encoder values for each frame of video so that the rendering aligns with both camera views simultaneously. Although this ground truth measurement is based on visual alignment so does not have a meaningful way of measuring

---

[1] `https://libcinder.org/`

[2] `http://research.intusurg.com/dvrkwiki/`

the error associated with it, it provides an effective upper bound on the accuracy of a visual tracking method which seeks alignment between a model and visual data. This means that the maximum accuracy in 3D tracking obtainable by a visual tracking method is achieved when model and image data align maximally, a task that the human eye is well suited to solving. Using this system we correct the ground truth of all of our datasets obtaining a translation in mm and rotation matrix between the instrument and camera coordinate system. To evaluate each degree of freedom, we decompose the rotation matrix into extrinsic Euler angles in the order $[r_z, r_y, r_x]$ where $r_{x,y,z}$ refers to a rotation around the x, y or z axis respectively [170].

For our analysis of instrument tracking we collect 7 ex-vivo datasets containing either one or two robotic instruments where 3D ground truth is recorded for each instrument. As the da Vinci system enforces the camera and instruments to move asynchronously, the sequences are captured with the camera stationary. We evaluate each component of our method separately to demonstrate the performance when using single and multiple region level set trackers. In addition to the quantitative analysis of the ex-vivo datasets, alongside each dataset we also perform qualitative analysis showing selected frames where the original camera image is shown alongside examples where the instrument is rendered at the current pose estimated by the tracker.

4.6.3 Ex-Vivo Experiments

We perform an analysis of the 3D pose tracking ability of the multiple region level set method we have proposed and compare with the more standard single region level set found in many 3D pose tracking methods in the computer vision literature [110, 140, 163]. The numerical results, summarized in Table 4.3, show that the multiple region level set achieves better performance than single region level sets in all degrees of freedom except the roll or r_x rotation, in particular scoring much better in the t_x direction and r_z direction. The higher performance in the t_x DOF is understandable as this constraint mainly acts in this direction, as the instruments are usually close to parallel to this axis and the constraint runs perpendicular to this axis. The individual datasets are analysed with trajectory plots (see Figure 4.12 - 4.29) showing the $[X, Y, Z]^T$ position and rotation in 3D space where the results for a single region (SR) level set tracker, shown in red, multiple region (MR) level set tracker, shown in blue, are compared with the hand corrected kinematic ground truth, shown in pink. Datasets 1 and 2 show interesting cases when the multiple region level set tracker's superiority is demonstrated. As seen in the trajectory plot for dataset 1 in Figure 4.12 and in the qualitative results in Figure 4.13 the multiple region tracker gives considerably better results where it maintains a position much closer to the true instrument tip when it is occluded behind the tissue. In dataset 2, where the trajectory plot is shown in Figures 4.14 and 4.15 with qualitative analysis in Figures 4.16 and 4.17, the extra constraints provided by the divide between the 2 regions prevents the tracker from failing when the instruments occlude one another. The other dataset where interesting results are observed are in dataset 6 where the single region tracker incorrectly rotates out of plane into a pose which is roughly consistent with the silhouette but grossly inconsistent with the image data. However, the constraint between the two regions prevents the multiple region tracker from making the same mistakes. The remaining datasets have roughly similar performance except in the case of roll rotation which appears to be regularly inconsistent with multiple regions and accounts for the majority of the cases where the single region level set is superior. However, as we observe that this is a general problem with both tracking systems we intend to solve it with the interior point tracker which is validated in the next section.

Figure 4.11: The setup in our lab showing the DVRK control box attached to a classic da Vinci with a stereo laparoscope. This control box connects to the MTMs and the PSMs allowing complete control of the PSMs using the MTMs in a master-slave configuration or alternatively the arms can be positioned using a software control system. This system is connected to a PC where frame data is captured using a Blackmagic Decklink Quad ® SDI capture card with an NVidia Quadro K4000 ® graphics card. This video data is synchronized to the joint kinematics which are collected using a robot operating system (ROS) interface.

Dataset	$t_x(mm)$	$t_y(mm)$	$t_z(mm)$	$r_x(rads)$	$r_y(rads)$	$r_z(rads)$
Dataset 1 - MR	2.37 ± 2.47	2.75 ± 2.68	1.72 ± 1.29	0.41 ± 0.24	0.22 ± 0.05	0.33 ± 0.07
Dataset 2 i - MR	2.54 ± 2.35	1.08 ± 0.87	10.10 ± 7.68	1.59 ± 1.60	0.17 ± 0.10	0.13 ± 0.08
Dataset 2 ii - MR	1.12 ± 0.71	2.31 ± 1.08	4.98 ± 3.77	1.09 ± 0.62	0.13 ± 0.05	0.22 ± 0.05
Dataset 3 - MR	3.19 ± 4.32	3.06 ± 2.35	8.94 ± 7.16	1.99 ± 1.61	0.20 ± 0.11	0.12 ± 0.11
Dataset 4 - MR	0.89 ± 0.98	0.65 ± 0.55	2.13 ± 1.69	1.04 ± 0.31	0.06 ± 0.07	0.08 ± 0.05
Dataset 5 - MR	3.66 ± 2.06	2.78 ± 1.63	12.76 ± 5.57	0.20 ± 0.19	0.31 ± 0.14	0.21 ± 0.09
Dataset 6 i - MR	0.90 ± 0.66	0.75 ± 0.58	5.08 ± 3.73	2.63 ± 1.48	0.08 ± 0.05	0.03 ± 0.02
Dataset 6 ii - MR	1.90 ± 0.43	1.12 ± 0.13	11.01 ± 1.62	0.08 ± 0.07	0.05 ± 0.02	0.06 ± 0.02
Dataset 7 i - MR	0.85 ± 0.46	0.47 ± 0.31	2.12 ± 1.27	1.85 ± 1.50	0.16 ± 0.06	0.04 ± 0.03
Dataset 7 ii - MR	0.55 ± 0.31	0.52 ± 0.50	2.59 ± 2.21	0.18 ± 0.09	0.04 ± 0.02	0.12 ± 0.08

Table 4.1: Errors for 3D pose estimation for region only trackers using multiple regions (MR). The translation and rotation errors for each dataset are shown where datasets with two instruments are shown separately as Dataset n i or Dataset n ii for the left and right instrument respectively. The values shown are the mean error over all frames ± the standard deviation.

Dataset	$t_x(mm)$	$t_y(mm)$	$t_z(mm)$	$r_x(rads)$	$r_y(rads)$	$r_z(rads)$
Dataset 1 - SR	3.90 ± 2.74	4.53 ± 3.77	7.39 ± 5.30	0.36 ± 0.24	0.19 ± 0.07	0.29 ± 0.11
Dataset 2 i - SR	15.52 ± 12.84	3.60 ± 2.25	25.92 ± 17.04	1.23 ± 0.40	0.48 ± 0.27	1.50 ± 1.16
Dataset 2 ii - SR	31.61 ± 19.43	4.60 ± 4.01	11.26 ± 6.72	0.78 ± 0.61	0.45 ± 0.26	1.32 ± 0.87
Dataset 3 - SR	5.99 ± 4.42	2.49 ± 1.81	7.65 ± 5.94	0.45 ± 0.41	0.32 ± 0.17	0.25 ± 0.14
Dataset 4 - SR	1.10 ± 0.82	0.76 ± 0.65	2.65 ± 1.99	0.32 ± 0.17	0.07 ± 0.05	0.07 ± 0.05
Dataset 5 - SR	5.26 ± 1.72	2.74 ± 1.42	4.17 ± 3.71	0.22 ± 0.17	0.22 ± 0.17	0.16 ± 0.10
Dataset 6 i - SR	2.10 ± 1.56	0.64 ± 0.49	4.84 ± 3.16	0.85 ± 0.45	0.09 ± 0.06	0.03 ± 0.02
Dataset 6 ii - SR	1.93 ± 1.27	0.25 ± 0.21	3.79 ± 3.43	2.35 ± 0.60	0.12 ± 0.06	1.63 ± 0.42
Dataset 7 i - SR	0.94 ± 0.50	0.45 ± 0.28	1.99 ± 1.24	0.13 ± 0.10	0.13 ± 0.05	0.02 ± 0.01
Dataset 7 ii - SR	0.98 ± 0.80	0.59 ± 0.46	2.89 ± 2.23	0.14 ± 0.09	0.03 ± 0.03	0.12 ± 0.08

Table 4.2: Errors for 3D pose estimation for region only trackers using single regions (SR). The terms in this table are the same as Table 4.1.

	$t_x(mm)$	$t_y(mm)$	$t_z(mm)$	$r_x(rads)$	$r_y(rads)$	$r_z(rads)$
Mean error MR	**1.80 ± 1.48**	**1.55 ± 1.07**	**6.14 ± 3.60**	1.10 ± 0.77	**0.14 ± 0.07**	**0.13 ± 0.06**
Mean error SR	6.93 ± 4.61	2.07 ± 1.54	7.26 ± 5.08	**0.68 ± 0.32**	0.21 ± 0.12	0.54 ± 0.30

Table 4.3: Overall errors of 3D pose estimation for region only trackers using single region (SR) and multiple regions (MR) over all datasets. The values shown are the mean error over all frames ± the standard deviation. The lower values are shown in bold where t_x, t_y, t_z and r_z is lower for the MR tracker and r_x and r_y are lower for the SR tracker.

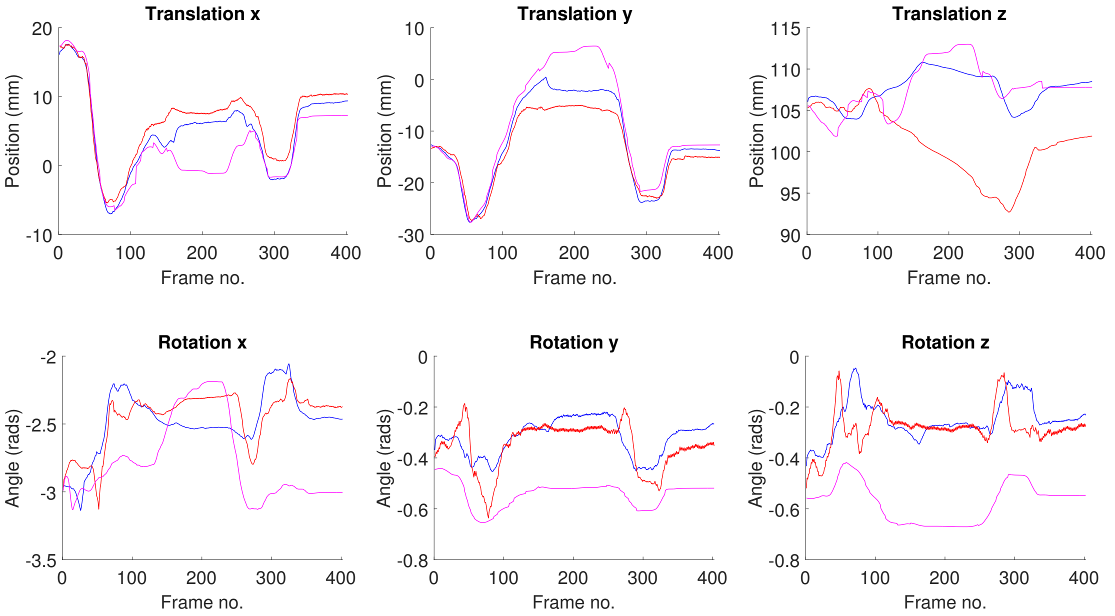


Figure 4.12: SR Tracker, MR Tracker, Ground Truth. The analysis of the translation and rotation of ex-vivo dataset 1 shows the improved performance of using multiple regions. Between frames 150 and 300 the instrument is partially occluded at the tip by tissue which causes the single region level set to translate along the shaft away from the occlusion. This is because without the information from the divide between the shaft and head, which is still visible, there is no constraint to prevent this from happening.

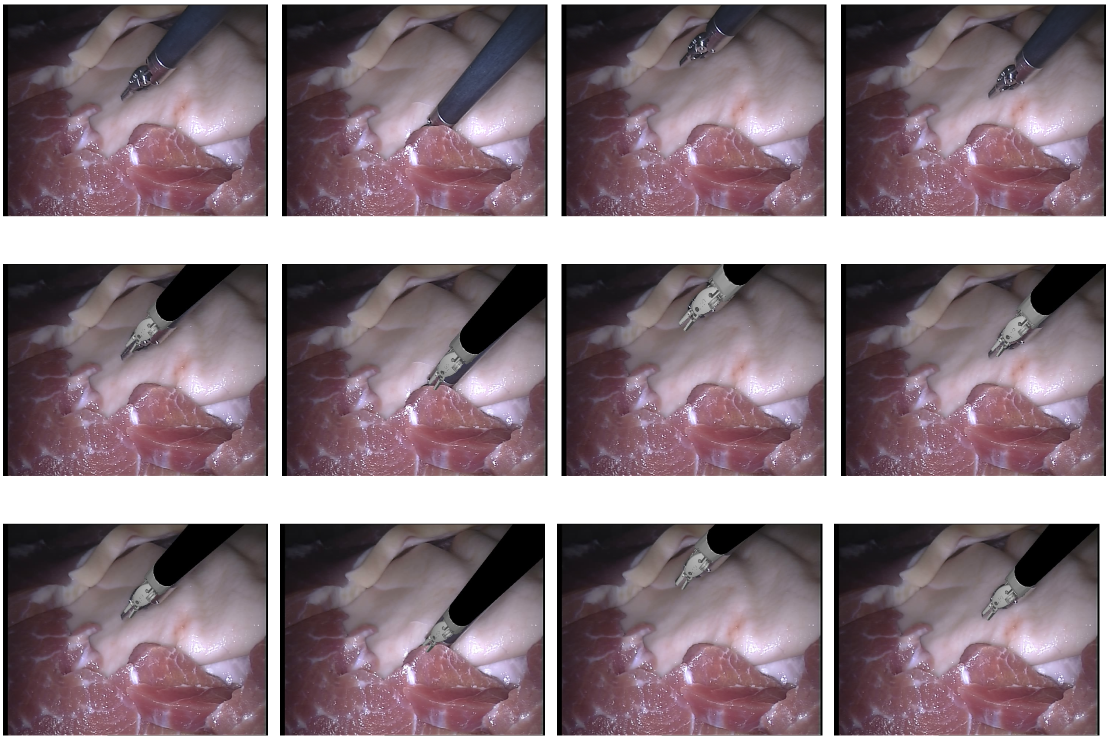


Figure 4.13: Sampled frames 100, 200, 300 and 350 from dataset 1 are shown in the top row and the corresponding frames when using a SR tracker are shown in the middle row and for the MR tracker in bottom row.

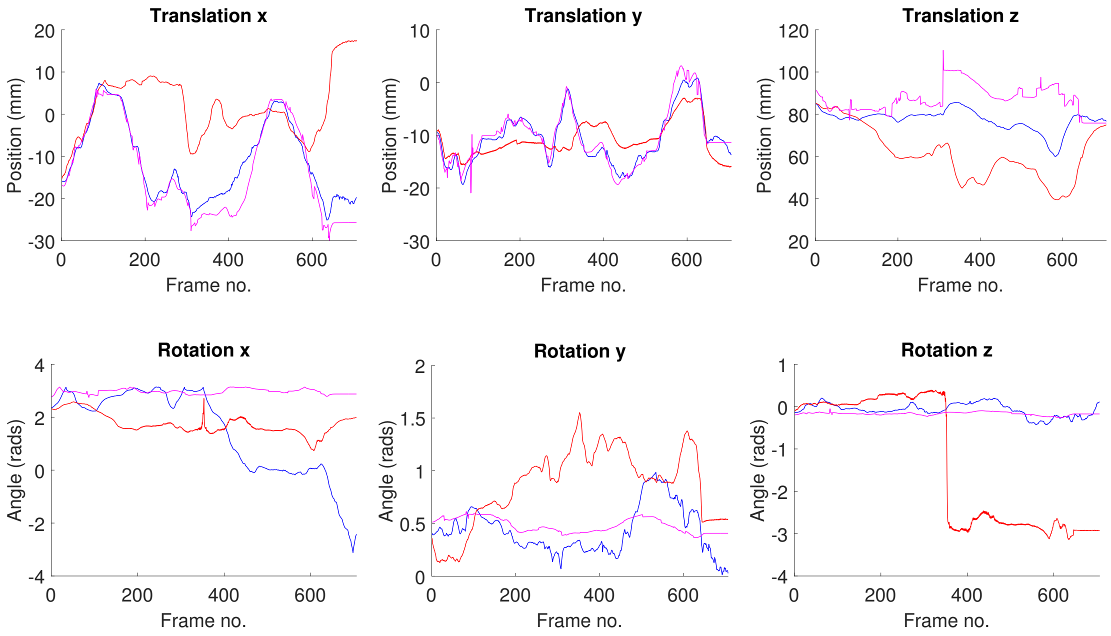


Figure 4.14: SR Tracker, MR Tracker, Ground Truth. Trajectory plots for the left instrument for ex-vivo dataset 2 using single and multiple regions. There is large visual occlusion when the left instrument passes over the right, particularly in t_x and r_z. The left instrument moves onto the region of the image where the right instrument lies and fails to recover when using a SR tracker. This is because the single color model has no constraint to prevent the model shifting along the shaft.

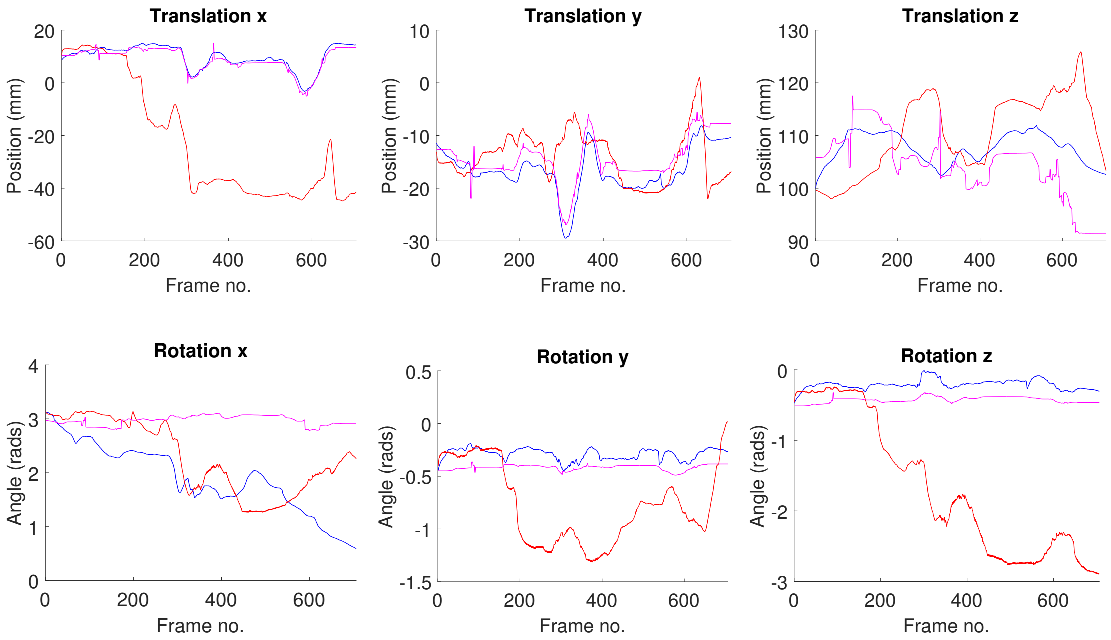


Figure 4.15: SR Tracker, MR Tracker, Ground Truth. Trajectory plots for the the right instrument for ex-vivo dataset 2 using single and multiple regions. Although errors are also quite large in the multiple region tracker compared with other datasets, it manages to track through this sequence.

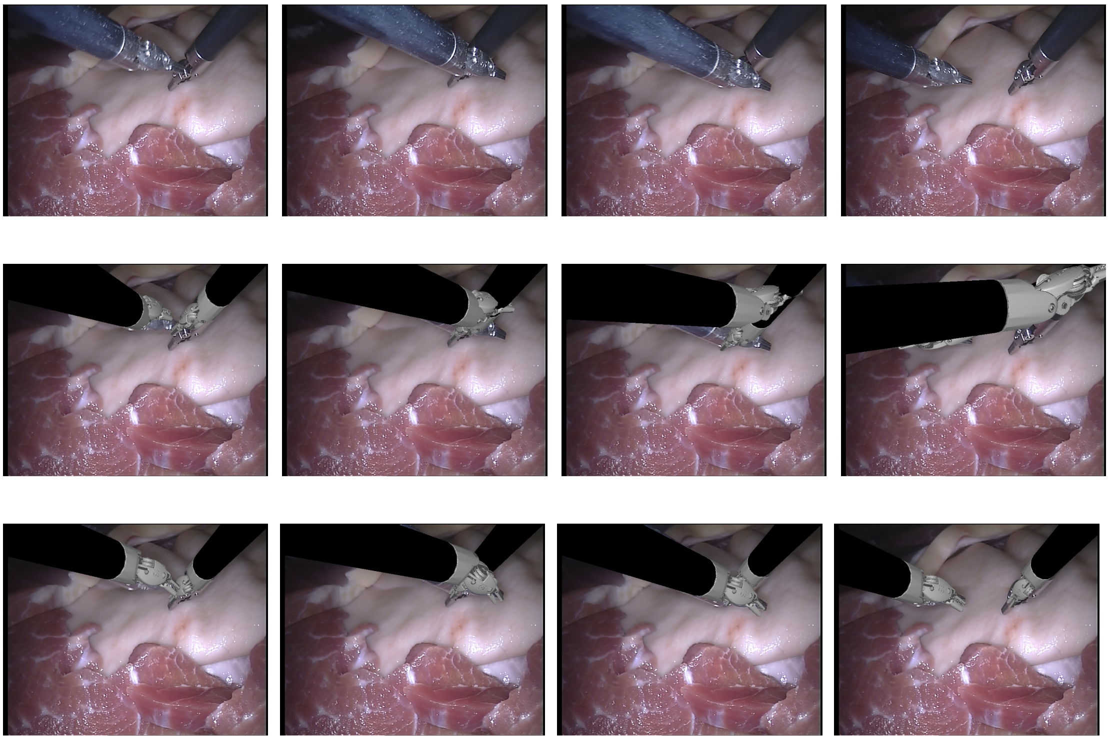


Figure 4.16: Qualitative analysis from ex-vivo dataset 2 showing the original frames 50, 100, 150, 250 in row 1, the corresponding frames from the SR tracker in row 2 and from the MR tracker in row 3. As the instruments begin to occlude one another, the SR tracker begins to fail.

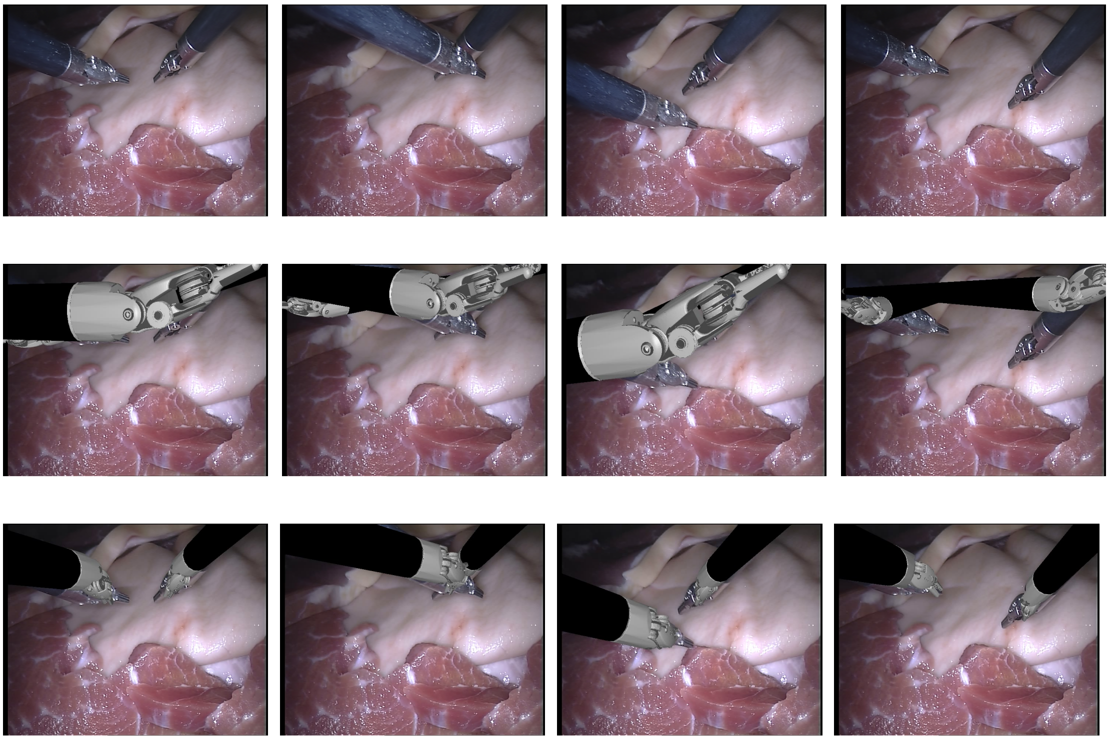


Figure 4.17: Qualitative analysis from ex-vivo dataset 2 showing the original frames 400, 500, 600 and 700 in row 1, the corresponding frames from the SR tracker in row 2 and from the MR tracker in row 3. Total tracking failure is observed in the SR tracker while the MR tracker successfully completes the sequence.

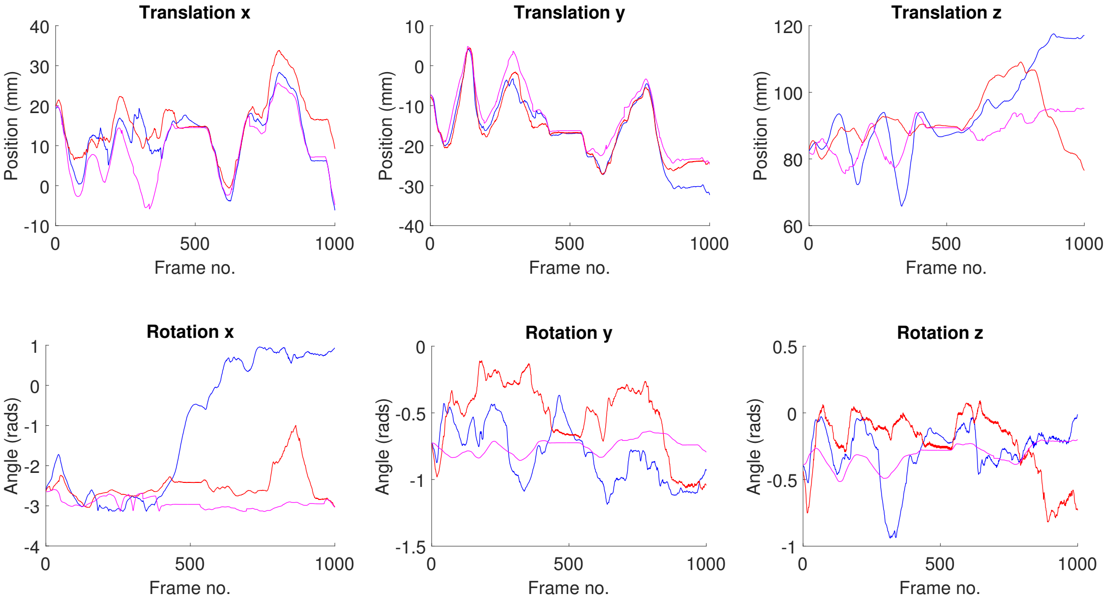


Figure 4.18: SR Tracker, MR Tracker, Ground Truth. The analysis of the translation and rotation of ex-vivo dataset 3 shows large errors when using both the SR and MR tracker. There are large r_x errors for the MR tracker when it rolls around the symmetry axis of the shaft. There are also larger rotation errors for both trackers during sequences of the video when the instrument shaft, which provides most of the rotational constraints, is mostly out of view.

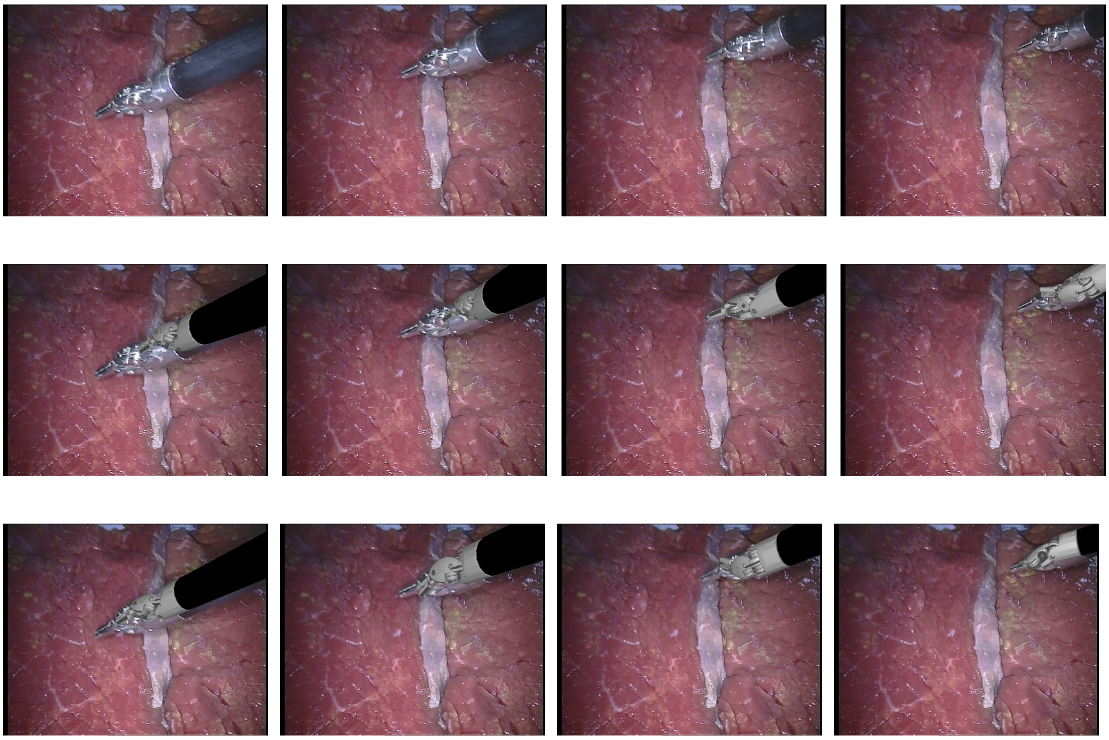


Figure 4.19: Sampled frames 100, 200, 550 and 850 from dataset 3 are shown in the top row and the corresponding frames when using a SR tracker are shown in the middle row and for the MR tracker in the bottom row.

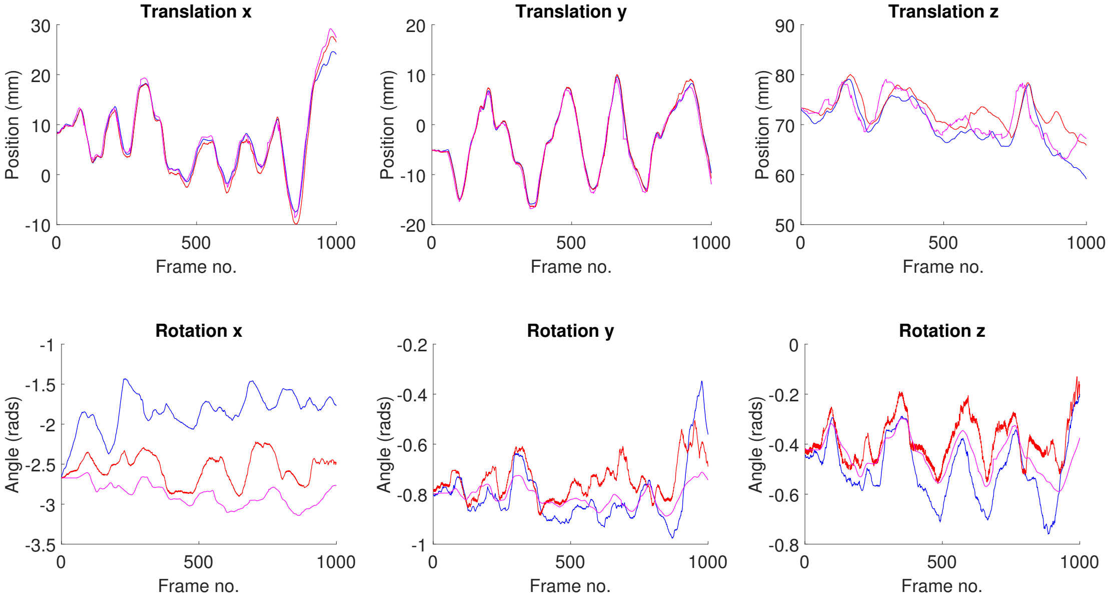


Figure 4.20: SR Tracker, MR Tracker, Ground Truth. The analysis of the translation and rotation of ex-vivo dataset 4 shows good accuracy in both the SR and MR trackers however the MR tracker exhibits large roll (r_x) error of around $\frac{\pi}{2}$ rads. The relative higher performance in this data is due to combination of a very clean classification and visibility of large parts of the instrument shaft.

Figure 4.21: Sampled frames 100, 200, 550 and 850 for dataset 4 are shown in the top row and the corresponding frames when using a SR tracker are shown in the middle row and the for the MR tracker in the bottom row. The multiple region tracker exhibits errors when it incorrectly rolls on its axis, as seen in the trajectory plots in Figure 4.20.

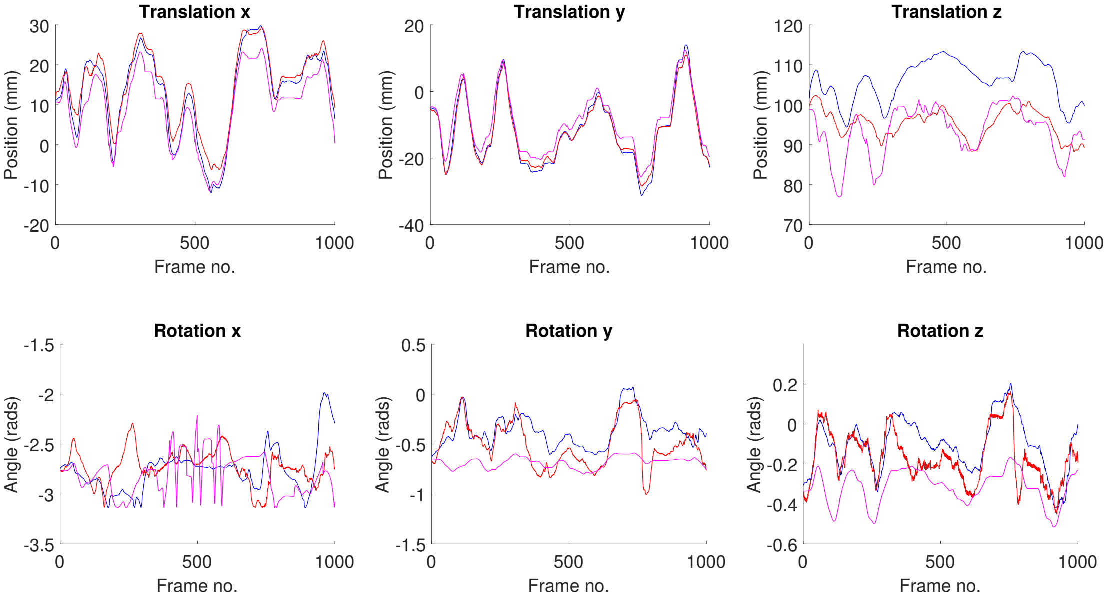


Figure 4.22: SR Tracker, MR Tracker, Ground Truth. The analysis of the translation and rotation of ex-vivo dataset 5 shows larger errors in both single and multiple region level set trackers, particularly in the z rotation of the multiple region tracker. This sequence is particularly challenging due to the larger t_z translation of around 90-100 mm compared with 70-80 mm in other datasets. Additionally, the lighter pink color in the background is often confused for the metallic instrument clasper by the RF.

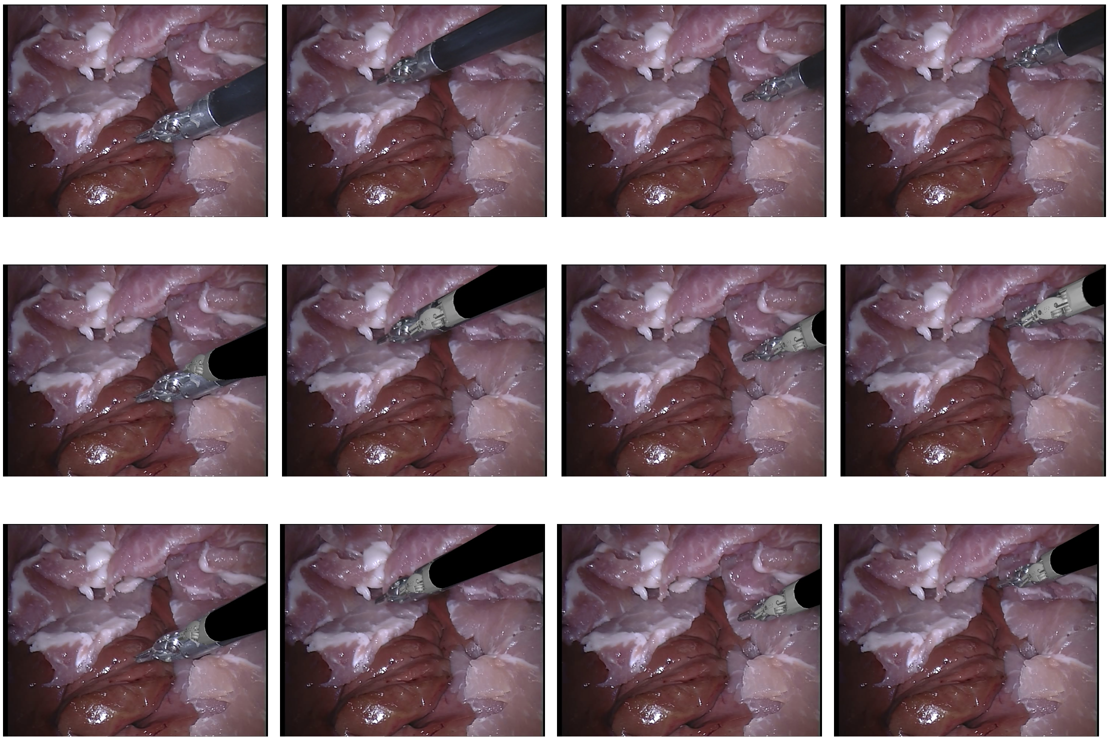


Figure 4.23: Sampled frames 100, 200, 550 and 850 for dataset 5 are shown in the top row and the corresponding frames when using a SR tracker are shown in the middle row and the for the MR tracker in the bottom row. The multiple region tracker correctly locks onto the divide between the plastic and metal but both fail to track most of r_x.

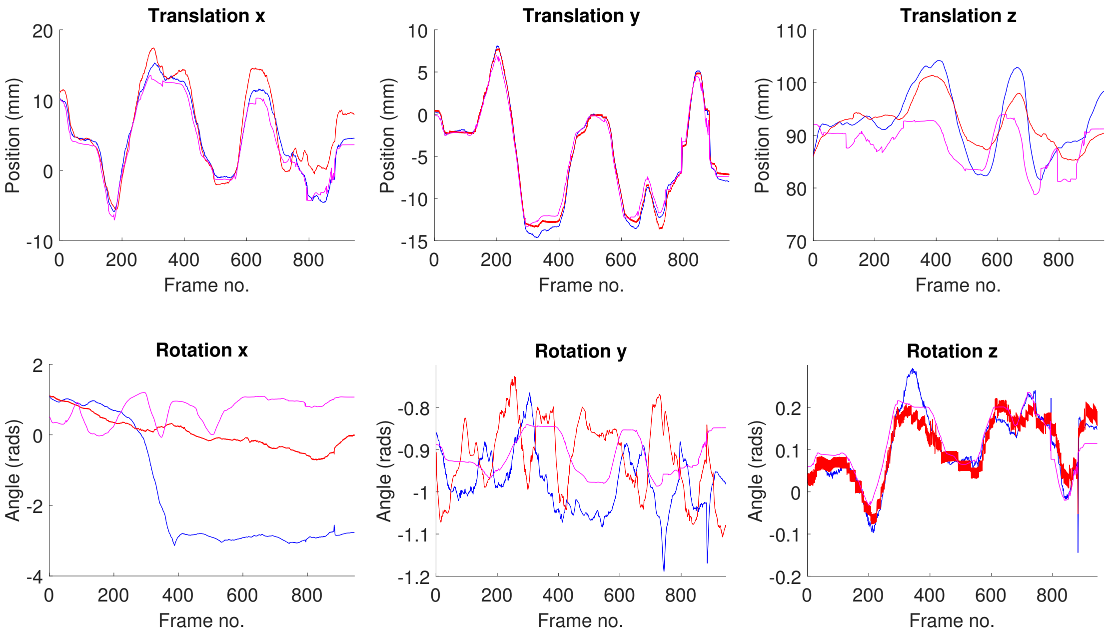


Figure 4.24: SR Tracker, MR Tracker, Ground Truth. Quantitative analysis of the right instrument for ex-vivo dataset 6. When tracking the right instrument, the multiple region level set tracker exhibits some r_x (roll) error again where it rotates π radians to a symmetric solution where the silhouette is identical. The instrument does not move over larger distances in r_y and r_z, which results in the comparatively noisy appearance of the plots, whereas the typical error is actually quite low (< 0.1 rads) in this sequence.

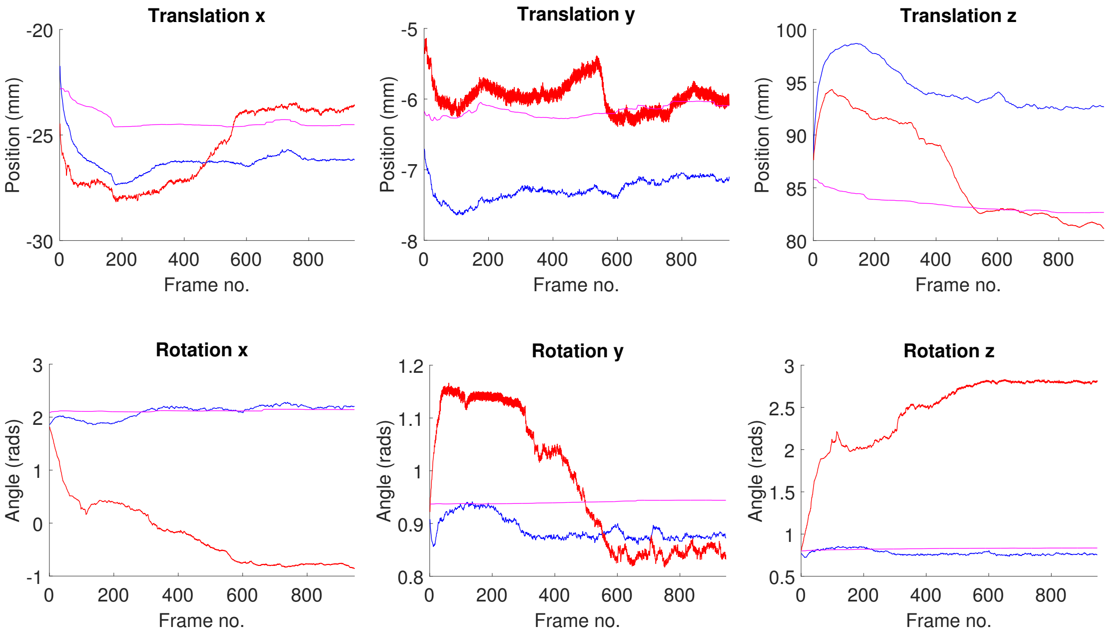


Figure 4.25: SR Tracker, MR Tracker, Ground Truth. The trajectories for the left instrument for ex-vivo dataset 6 using the SR and MR trackers. Inaccuracies are observed when tracking with the SR tracker, which exhibits a limitation of the silhouette only model. A large yaw rotation is observed, which produces a very inaccurate pose, despite the fact the silhouette is quite accurately tracked. The MR tracker does not have this problem as the contour between the two regions constrains this rotation.

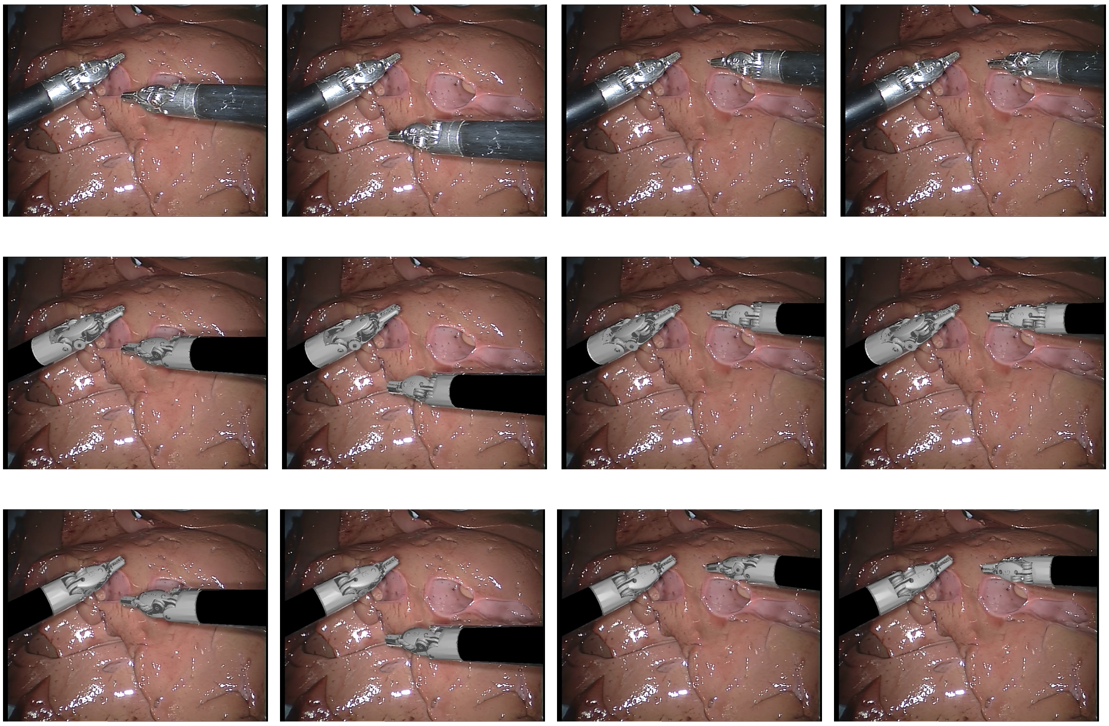


Figure 4.26: Qualitative analysis from ex-vivo dataset 6 showing frames 100, 200, 350 and 400. Row 1 shows the original frames, row 2 shows the SR tracker and row 3 shows the MR tracker. The left instrument when using the SR tracker begins to rotate away from the correct pose while maintaining the correct silhouette.

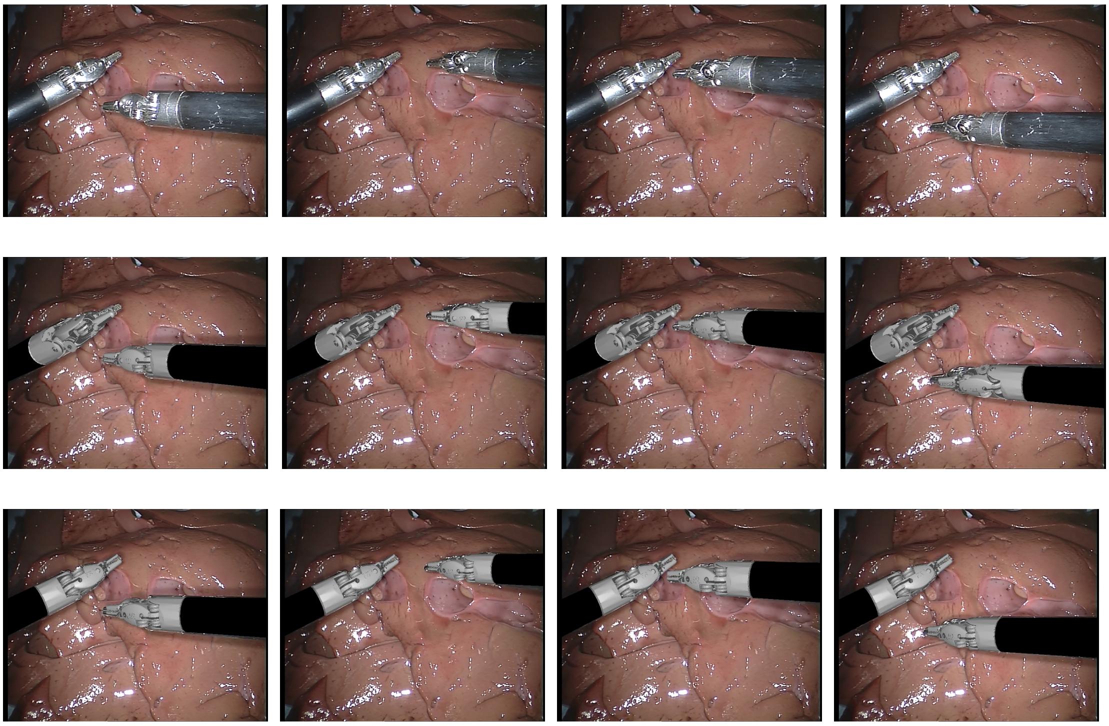


Figure 4.27: Qualitative analysis from ex-vivo dataset 6 showing frames 500, 650, 750, and 850. Row 1 shows the original frames, row 2 shows the SR tracker and row 3 shows the MR tracker. The frames show significant misalignment between the SR level set estimate and the instrument is clearly visible as the left instrument rotates far from the correct pose.

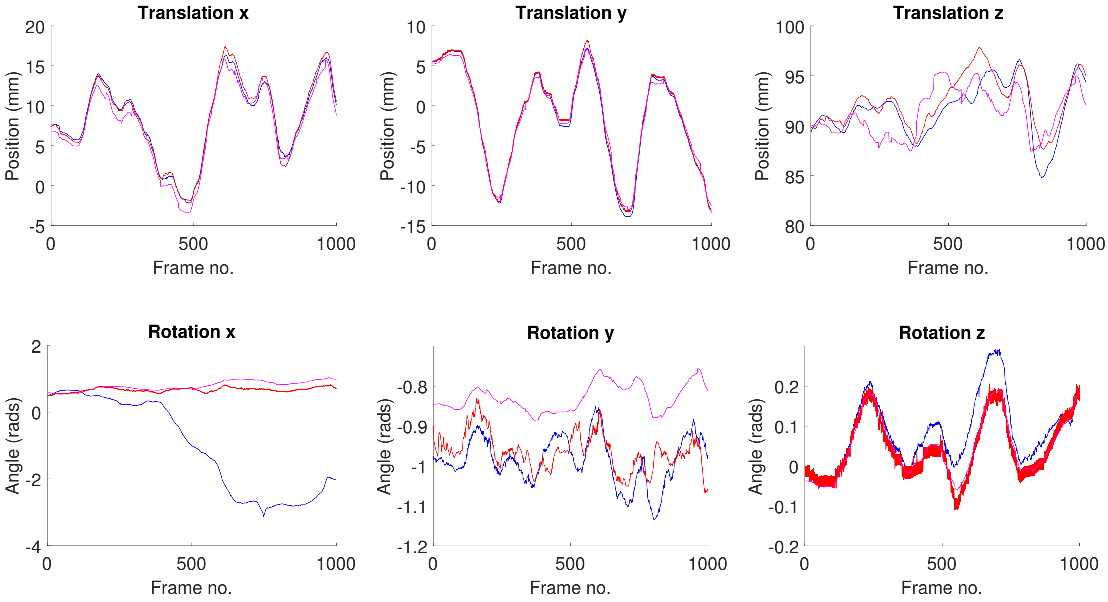


Figure 4.28: SR Tracker, MR Tracker, Ground Truth. Quantitative analysis of the right instrument for ex-vivo dataset 7 for the right instrument where the tracking is mostly accurate but the instrument exhibits r_x (roll) rotation errors.

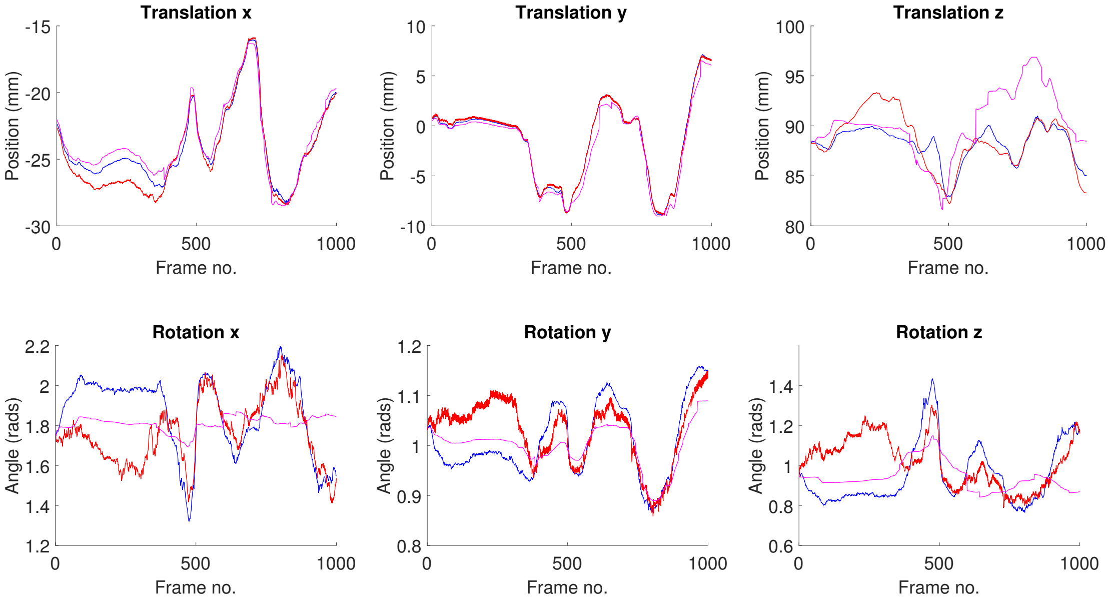


Figure 4.29: SR Tracker, MR Tracker, Ground Truth. The analysis of the translation and rotation of of the left instrument for ex-vivo dataset 7 with SR and MR trackers which both obtain very accurate results for this instrument.

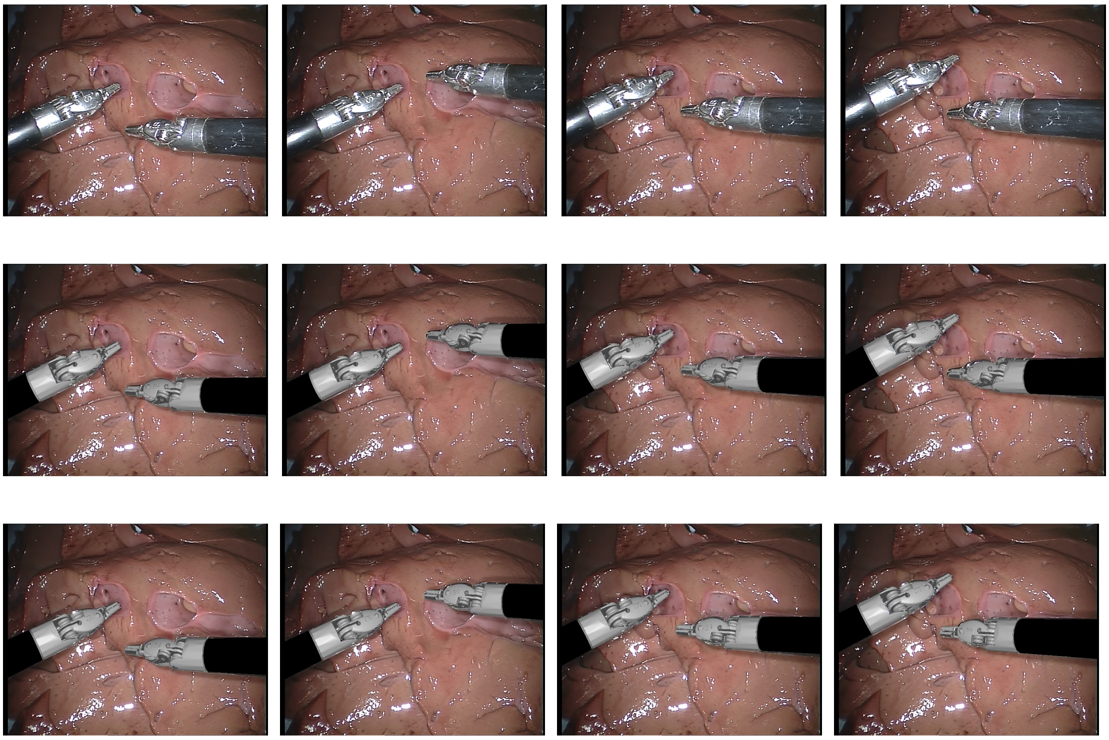


Figure 4.30: Qualitative analysis from ex-vivo dataset 7 showing frames 100, 200, 350, and 400. Row 1 shows the original frames, row 1 shows the single region tracker and row 2 shows the multiple region tracker. Both trackers are quite accurate over this sequence, as can be seen from the good visual alignment.

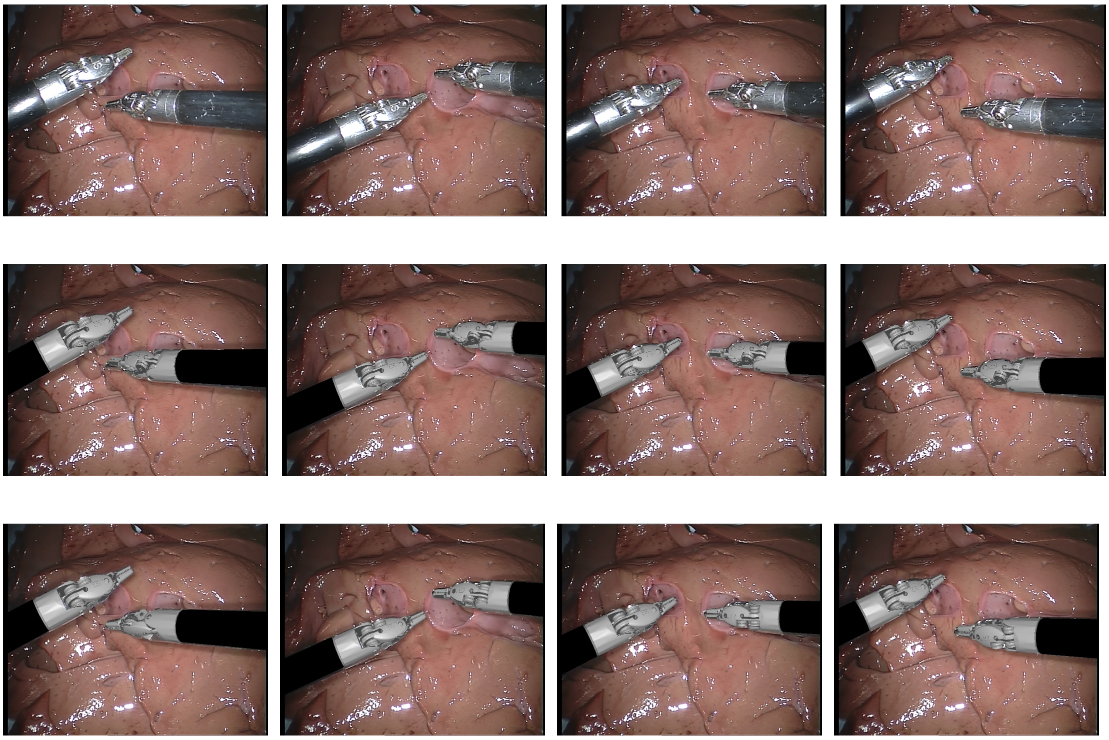


Figure 4.31: Qualitative analysis from ex-vivo dataset 7 showing frames 500, 650, 750, and 850. Row 1 shows the original frames, row 1 shows the single region tracker and row 2 shows the multiple region tracker. The R_x error in the multiple region tracker is visible in frame 750 and 850.

4.7 Conclusion

In this chapter we demonstrate a novel system for recovering the 3D pose of instruments in surgical images. The SR and MR level set based methods both demonstrate that they can recover the translation and r_y and r_z rotation angles quite accurately, but as shown in Table 4.3, the accuracy of the MR method is better, particularly in the t_x DOF and also in the r_y DOF. The likely cause of the improvements in the t_x direction are due to the constraint provided by the shaft and clasper divide mostly acting in this direction for instruments which are close to parallel to the x axis of the camera frame $\mathcal{F}_{cam}$. Additionally, the rotation around the y axis of the camera is also improved by this vertical constraint. However, due to ambiguities in the shape of the instruments, the silhouette alone is not a strong cue to recover the roll rotation (r_x) and this particularly seems to affect the MR tracker. This is noticeable particularly in dataset 3, where the trajectory plots are shown in Figure 4.19 and the right instrument of dataset 6, where the trajectory plots are shown in Figure 4.25.

Interesting improvements from the MR tracker are visible particularly when there are occlusions of parts of the instrument when the extra constraint created by the color change between the metal and plastic parts of the instrument prevent it from translating along the shaft in either direction. The occlusion of the tip can often be a particular challenge when working with surgical instruments, as the distal end of the instrument is fully occluded by the edge of the frame. This presents a particular challenge as the pixel labeling provides no useful information about where to move the instrument, as shown in datasets 1 and 2 with trajectory plots shown in Figure 4.12, 4.14, 4.15. An additional interesting case occurs in dataset 6, where Figure 4.26 and Figure 4.27 show the trajectory plots, where the SR tracker rotates out of alignment with the ground truth, while still maintaining a reasonably valid silhouette. However, using the MR tracker prevents this situation from occurring as the boundary between the shaft and clasper provides a constraint against this rotation.

One of the major limitations of the method involves dealing with situations when the silhouette computed by the RF classified is not informative enough to lead to good pose estimation. This can occur when the classification is noisy leading to poorly defined silhouettes or alternatively when the silhouette is ambiguous due to symmetries in the instrument shape. This can in principle be solved by adding additional information that is less dependent on simple features such as color which are easily confused in the case of poor lighting. In Chapter 5 we will aim to solve this problem by experimenting with additional features which we hope will be able to assist the pose estimation in cases where the color features fail. An additional limitation of this technique it that it is fully rigid and therefore cannot capture the articulations of the robotic instruments. In addition to this being a general limitation in using the technique in clinical applications, it also limits that accuracy in these sequences as although the instruments are held in a rigid pose, inevitable movements of the user's hand introduces small modifications which greatly impacts the ability of the shape based tracking system to converge. This problem will be addressed in Chapter 6. A final major limitation of our method is the computational runtime. Although our method is highly unoptimized C++ which was written for prototyping and experimentation rather than performance, there is still considerable concern over the runtime which can amount to 1-2 seconds per gradient descent step with between 15-25 steps used for each frame for convergence. There are however, several methods available to speed up the processing. Firstly, fully utilizing GPU programming has demonstrated real-time performance on similar methods as the cost function is evaluated as a independent sum-over-pixels [110]. Additionally, we use a fixed number of steps for convergence, which means there is considerable redundant processing on the majority of frames when the minimum is reached quickly. However, accurately detecting when the minimum is reached proved challenging meaning we did not include a system for convergence detection in our method, instead focusing on providing accuracy rather than speed.

Chapter 5

Incorporating Sparse Features for 3D Pose Estimation

5.1 Introduction

Despite the significant advantage to the global region based features and their well documented strength in solving 3D pose estimation problems they have several limitations. One clear challenge occurs when several poses correspond to very similar silhouettes which is often compounded when noise occurs around the contour. For some object types, particularly those with a symmetry axis such as cylinders, this problem is particularly challenging and this was routinely observed for surgical instruments during evaluation in Chapter 4. Unarticulated surgical instruments are close to cylindrical (see Figure 5.1) and the shape misalignment cues used to drive the region based pose estimation are much weaker for this degree of freedom. One way to account for the errors in the silhouette is to integrate some fine grain local features on the body of the instrument. Using point matching to estimate pose has been extensively studied [171, 172] but in the case of surgical instruments the relatively featureless surfaces mean that few methods have achieved success without a secondary source such as kinematics from robotic systems to disambiguate the matches [71].

The task of estimating 3D pose from a set of 2D-3D point correspondences is usually referred to as perspective-n-point (PnP) pose estimation [139] and in principle can be achieved with a unique solution from 4 or more point matches, assuming that the points are not coplanar and that no set of 3 points are collinear. The basic idea involves projecting N 3D vertex locations onto the camera image plane using Equation 4.4 and then finding matches for each of these projected points in the image. Methods from the computer vision literature can be divided between direct methods, which seek to estimate the parameters of ${}^{cam}\mathbf{T}_{model}$ by solving linear systems of equations using methods such as singular value decomposition (SVD) [173] or iterative methods, which incrementally solve a model-data alignment cost function until an appropriate minimum is found [174, 175].

In this chapter we address limitations in region based tracking frameworks with feature point tracking within the regions using SIFT and optical flow which we show both improves accuracy in cases when the region segmentation is poor and also aids in recovering the roll rotation of the instrument. Our methods are validated on the same calibrated ex-vivo data as the region based tracking method of Chapter 4 which demonstrates the benefits provided by adding this type of feature. The work presented in this chapter was described in the publications [141, 142, 176].

5.2 Tracking Surface Features

The idea of tracking 2D information on the instrument surface as an additional method of constraining the pose estimation is very simple and works on the principal that if we can match 2D information in the

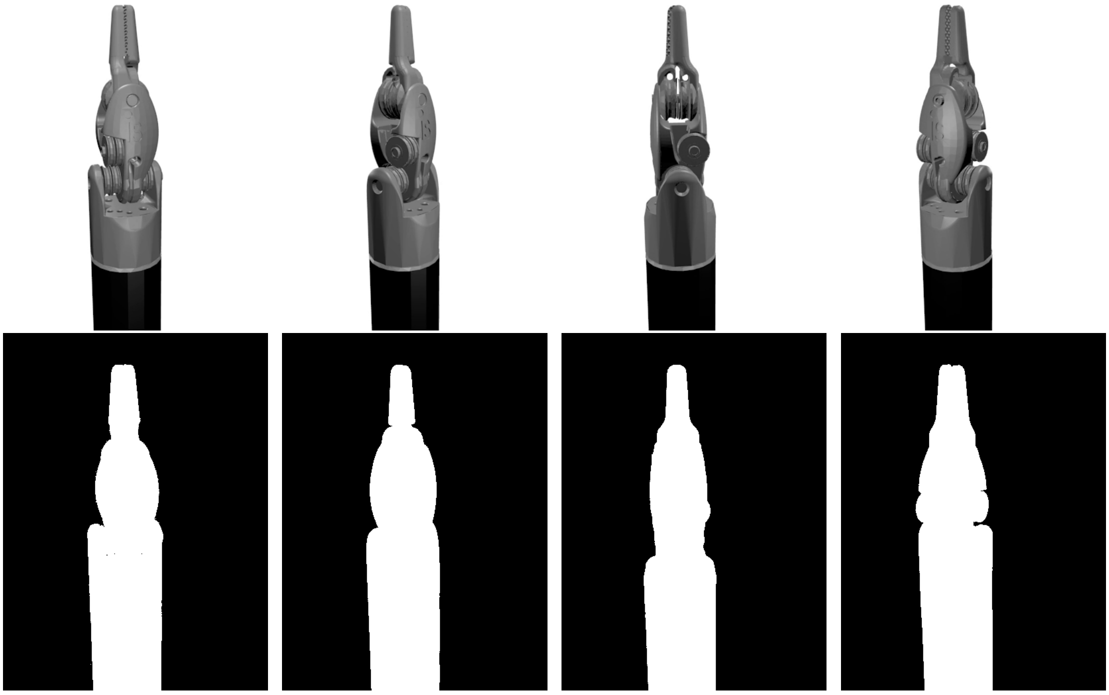


Figure 5.1: This illustrates one of the challenges in only using silhouette features when estimating pose of robotic instruments. As the da Vinci LND instrument rotates on its axis there is very limited change in its silhouette which will make tracking this DOF challenging.

image plane to 3D points on the model surface, we can estimate the 3D transformation to the instrument by minimizing the reprojection error between the predicted 2D point locations $[x, y]^T$ defined by the projection function in Equation 4.4 and their correspondences $[\hat{x}, \hat{y}]^T$ in the image. This can be defined by the objective function:

$$E_{point}(\boldsymbol{\theta}) = \sum_{i \in W^{t+1}} ||\mathbf{K}\mathbf{T}(\boldsymbol{\theta})\mathbf{X}_i^t - [\hat{x}_i^{t+1}, \hat{y}_i^{t+1}]^T||_2^2 \tag{5.1}$$

where $||.||_2^2$ denotes the squared L_2 norm, although other distance metrics are commonly used [172]. $[\hat{x}_i^{t+1}, \hat{y}_i^{t+1}]^T$ denotes a corresponding point location in the frame at time $t+1$ which was matched with the point projected from the vertex location $\mathbf{X}_i^t$ at t. W^{t+1} is the set of matched points between frames at times t and $t+1$.

To achieve this, an appearance model for the 2D features can be learned, which enables them to be localized in the image, using a matching method such as RFs [70]. However, a particular challenge in this case is achieving highly precise localizations as the instrument surface is relatively featureless and this means that large regions or external validation sources such as kinematics [10] must be used to obtain sufficiently accurate matches. However, as we have a course alignment of a 3D model from the region based level set method, we can use an alternative method which exploits small frame-to-frame motion to more reliably match points between frames t and $t+1$. This does not require a complex matching mechanism as spatial constraints means brute force matching becomes a viable strategy. By assuming that the pose in frame t is correct, the 3D transformation which produces the observed motion between the 2 frames is calculated and the pose is estimated incrementally from there. A major disadvantage of this method when compared to the appearance matching methods in the literature [70] is that it uses relative motion between frames, rather than a direct estimate of the full transform ${}^{cam}\mathbf{T}_{model}$ at each frame. This can lead to significant drift if used as a solitary feature for tracking as small errors

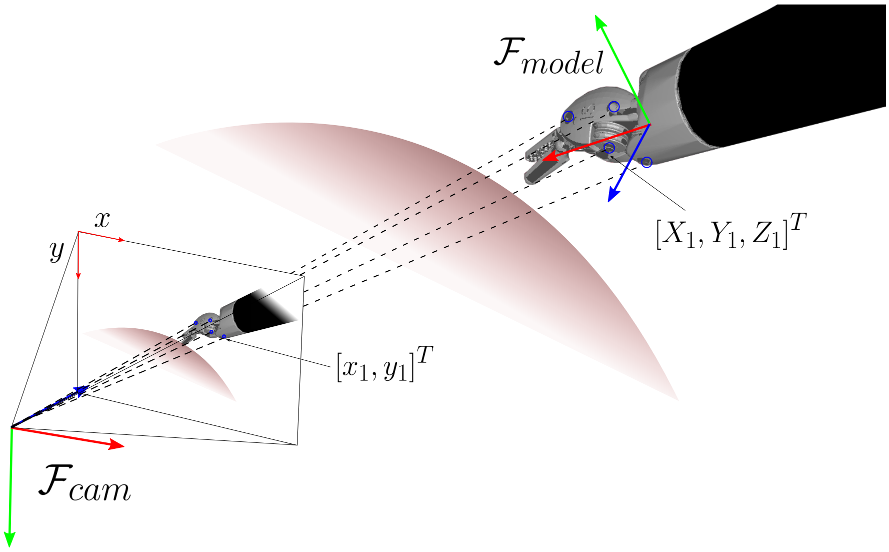
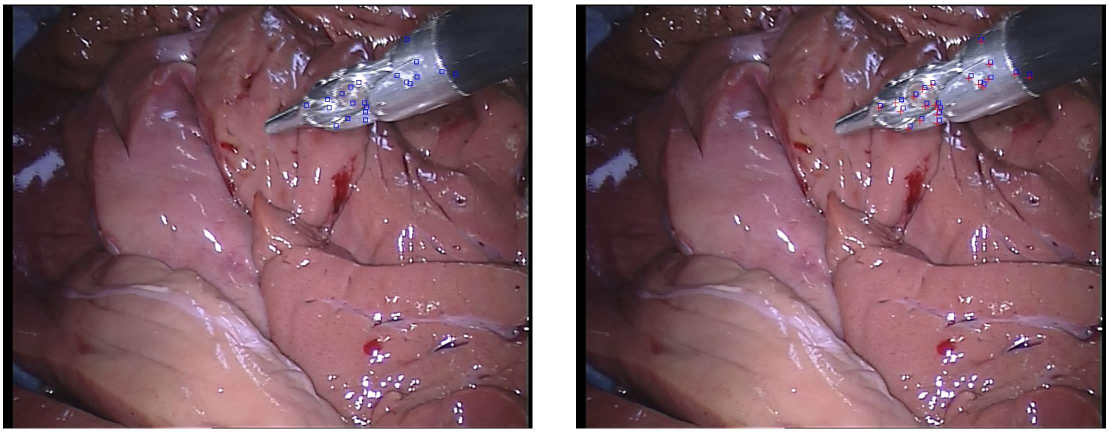


Figure 5.2: A set of 4 points $[X_i, Y_i, Z_i]^T$ defined in the model reference frame $\mathcal{F}_{model}$ are projected into the image using the transform ${}^{cam}\mathbf{T}_{model}$ which aligns each of them to their correspondences in the image plane $[x_i, y_i]^T$.

in each frame accumulate until tracking is lost. However, if combined with the region features, which are not as sensitive to drift, the two feature types can be combined in a way that minimizes their respective weaknesses. The idea of combining region features with motion based point features has been explored within the mainstream computer vision community using a classic infinite dimensional level set formulation which was combined with SIFT features and optical flow to track general 3D objects [140]. However, this implementation has proved extremely slow with up to 4 minutes required for optimization of each frame. To experiment with how 2D feature points can improve level-set based tracking of surgical instruments, we investigate SIFT features for this task, using the RootSIFT implementation [177] which provides better quality matching than vanilla SIFT and pyramidal optical flow [178].

5.2.1 Tracking Features with SIFT Matching


(a)

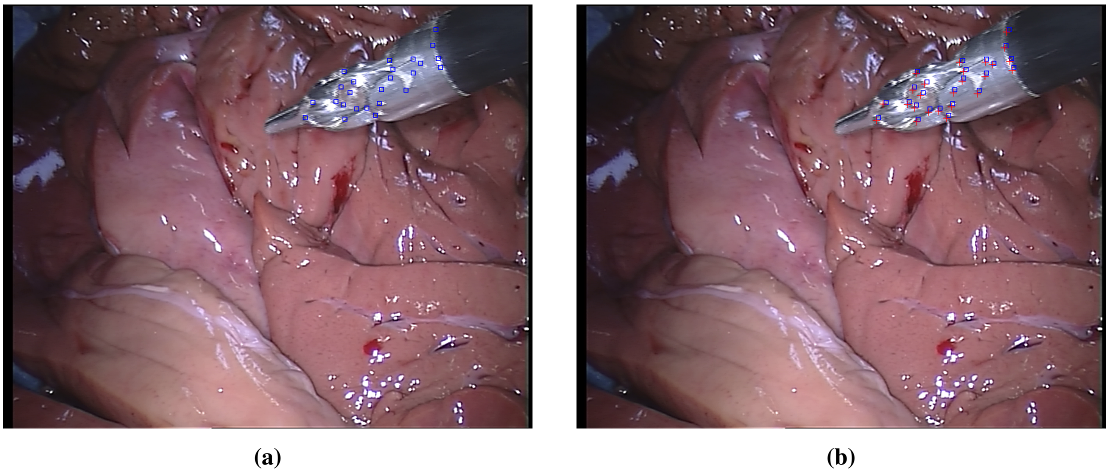
(b)

Figure 5.3: SIFT feature tracking between 2 frames. (a) The original features in the frame at time t denoted with red squares. (b) The frame at time $t + 1$ where matches between the features in frame t and this frame are shown as red crosses. Around 20 features are tracked, mostly around the instrument head where the most texture is present.

SIFT [179] is a popular technique for finding correspondences between interest points in multiple images taken with different spatial, temporal or illuminance conditions. The algorithm first applies a detector which extracts interest points that are stable across different scales by locating extrema in a multi-scale Difference of Gaussian (DoG) image which provides a lower computational cost version of the Laplacian of Gaussian (LoG) filter. The scale space is constructed with a pyramidal grouping known as octaves, where each octave is half the resolution of the octave that preceded it. Within each octave, the image is blurred with progressively larger Gaussian kernels and the idea is that any feature which persists across scales will appear as a local maximum within an octave. These local maxima are filtered for contrast and edges (which provide minimal matching power) to find a final set of good candidate points, known as keypoints. Using these detected feature locations, the SIFT descriptor is computed. To ensure that the feature is orientation invariant, a dominant orientation is computed which is used as a reference orientation when matching is performed. This is achieved by creating a histogram of the gradient orientations around the keypoint, where 10° bins are normally used. The gradient magnitude is used to weight each orientation's contribution to the bin and the maximum of the histogram is chosen as the dominant orientation. The SIFT feature itself is computed by sampling gradients on a 16×16 Gaussian weighted grid around the keypoint. This grid is further subdivided into 4×4 local neighborhoods where a new 45° bin gradient orientation histogram is computed for each neighborhood. These are then concatenated together to create a 128 dimensional vector which is normalized to enforce illumination invariance.

Few methods have managed to improve on the classic algorithm for matching. One of the notable successes was RootSIFT [177] which aimed to improve the traditional SIFT implementation's matching technique which uses Euclidean distance to compare feature vectors. This been shown to perform poorly when comparing histograms as larger bin values can dominate smaller ones so RootSIFT instead compared with the Hellinger distance, which gives a larger weighting to smaller bin values.

5.2.2 Tracking Features with Optical Flow



(a) (b)

Figure 5.4: Optical flow feature tracking between 2 frames. (a) The original features in the frame at time t denoted with red squares. (b) The frame at time $t + 1$ where matches between the features in frame t and this frame are shown as red crosses. More features are tracked with optical flow, compared with SIFT, particularly on the boundary between the shaft and the metal clasper.

Optical flow refers to the intensity changes observed in an image sequence due to the motion of objects in the image relative to the camera. It is centered on the *brightness consistency assumption* that assumes that the observed grayscale illuminance of a scene point observed by the camera is constant [180]. This is defined mathematically as:

$$I(x, y, t) = I(x + \delta x, y + \delta y, t + \delta t) \tag{5.2}$$

from which it follows that:

$$\frac{\partial I}{\partial x}V_x + \frac{\partial I}{\partial y}V_y = -\frac{\partial I}{\partial t} \tag{5.3}$$

under the assumption that movements are small. Given an image point $\mathbf{x}^t = [x^t, y^t]^T$ in an image at time t, the goal of optical flow is to find the location of $\mathbf{x}^{t+1} = [x^{t+1}, y^{t+1}]^T$ in a subsequence image at time $t + \delta t$ ($\delta t = 1$) where $\mathbf{x}^{t+1} = \mathbf{x}^t + \mathbf{V}$ and $\mathbf{V}$ is the optical flow at $\mathbf{x}^t$. The optical flow can be computed using many methods, one of the most popular is the Lucas-Kanade (LK) method which estimates the flow field under a locally affine model [181]. The basic assumption of the LK method is that the optical flow in the local neighborhood of a pixel is constant. By constructing a small window centered on a point $\mathbf{x}$, the constant optical flow for the window is constructed as linear least squares problem as:

$$\begin{aligned}
\frac{\partial I(\mathbf{x}_0^t)}{\partial x}V_x + \frac{\partial I(\mathbf{x}_0^t)}{\partial y}V_y &= -\frac{\partial I(\mathbf{x}_0^t)}{\partial t} \\
\frac{\partial I(\mathbf{x}_1^t)}{\partial x}V_x + \frac{\partial I(\mathbf{x}_1^t)}{\partial y}V_y &= -\frac{\partial I(\mathbf{x}_1^t)}{\partial t} \\
\frac{\partial I(\mathbf{x}_2^t)}{\partial x}V_x + \frac{\partial I(\mathbf{x}_2^t)}{\partial y}V_y &= -\frac{\partial I(\mathbf{x}_2^t)}{\partial t} \\
&\;\;\;\; ... \\
\frac{\partial I(\mathbf{x}_n^t)}{\partial x}V_x + \frac{\partial I(\mathbf{x}_n^t)}{\partial y}V_y &= -\frac{\partial I(\mathbf{x}_n^t)}{\partial t}
\end{aligned}$$

where n features are tracked at points $(\mathbf{x}_0^t, \mathbf{x}_1^t, ... , \mathbf{x}_n^t)$. To balance the challenges of achieving tracking that is both robust to larger inter-frame motion and changes in lighting while still providing sub-pixel accuracy the LK method is normally performed over several scales [178]. The highest resolution level of the pyramid is the capture resolution of the camera, and by successively halving the vertical and horizontal resolution of the image, coarse scale levels are obtained. The optical flow estimate is computed at the lowest resolution level and used as an initialization for the optical flow optimization at the next highest resolution level. This process is repeated until the optical flow in the full resolution images is computed.

5.2.3 Dealing with Interior Feature Errors

In both cases, when we track points we apply several constraints to ensure the matching is of high quality and that errors do not greatly affect the pose estimates. Firstly, we wish to prevent cases when occlusions between instruments prevent the surface features from one instrument being visible. To achieve this we use the same occlusion map that we use for the level set tracker and switch off the tracking of features when they are occluded by an instrument. Additionally, we switch off the tracking of a feature when it no longer is matched to a pixel inside the project contour of the tracked model and also apply a threshold to the maximum error in the feature reprojection of 25 pixels.

5.3 Optimization

We jointly optimize over the region based energy, referred to from here on as $E_{region}(\boldsymbol{\theta})$, and point based energy computed using either SIFT or optical flow, $E_{point}(\boldsymbol{\theta})$ using gradient descent and a weighting factor λ to allow both terms to have more equitable influence. In our experiments we set λ so that the Jacobians from the point estimates have 0.8 of the magnitude of the Jacobians from the region based energy:

$$E(\boldsymbol{\theta}) = E_{region}(\boldsymbol{\theta}) + \lambda E_{point}(\boldsymbol{\theta}) \tag{5.4}$$

where the derivative is computed as:

$$\frac{\partial E(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \frac{\partial E_{region}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} + \lambda \frac{\partial E_{point}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \tag{5.5}$$

$$\frac{\partial E_{point}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \sum_{i \in W_{t+1}} \frac{\partial}{\partial \boldsymbol{\theta}} ||\mathbf{K}\,{}^{cam}\mathbf{T}(\boldsymbol{\theta})_{model}\mathbf{X}_i^t - [\hat{x}_i^{t+1}, \hat{y}_i^{t+1}]||_2^2 \tag{5.6}$$

$$= \sum_{i \in W_{t+1}} 2\big[\mathbf{K}\,{}^{cam}\mathbf{T}(\boldsymbol{\theta})_{model}\mathbf{X}_i^t - [\hat{x}_i^{t+1}, \hat{y}_i^{t+1}]\big]\,\big[[\frac{\partial x_i^t}{\partial \boldsymbol{\theta}}, \frac{\partial y_i^t}{\partial \boldsymbol{\theta}}]^T - [\hat{x}_i^{t+1}, \hat{y}_i^{t+1}]^T\big]\big] \tag{5.7}$$

where the Jacobians of the vertex position with respect to the pose parameters $\boldsymbol{\theta}$ are given by the same terms as in the region cost in Equation 4.22 and Equation 4.23. We again optimize this cost with gradient descent using the same step size as in Chapter 4.

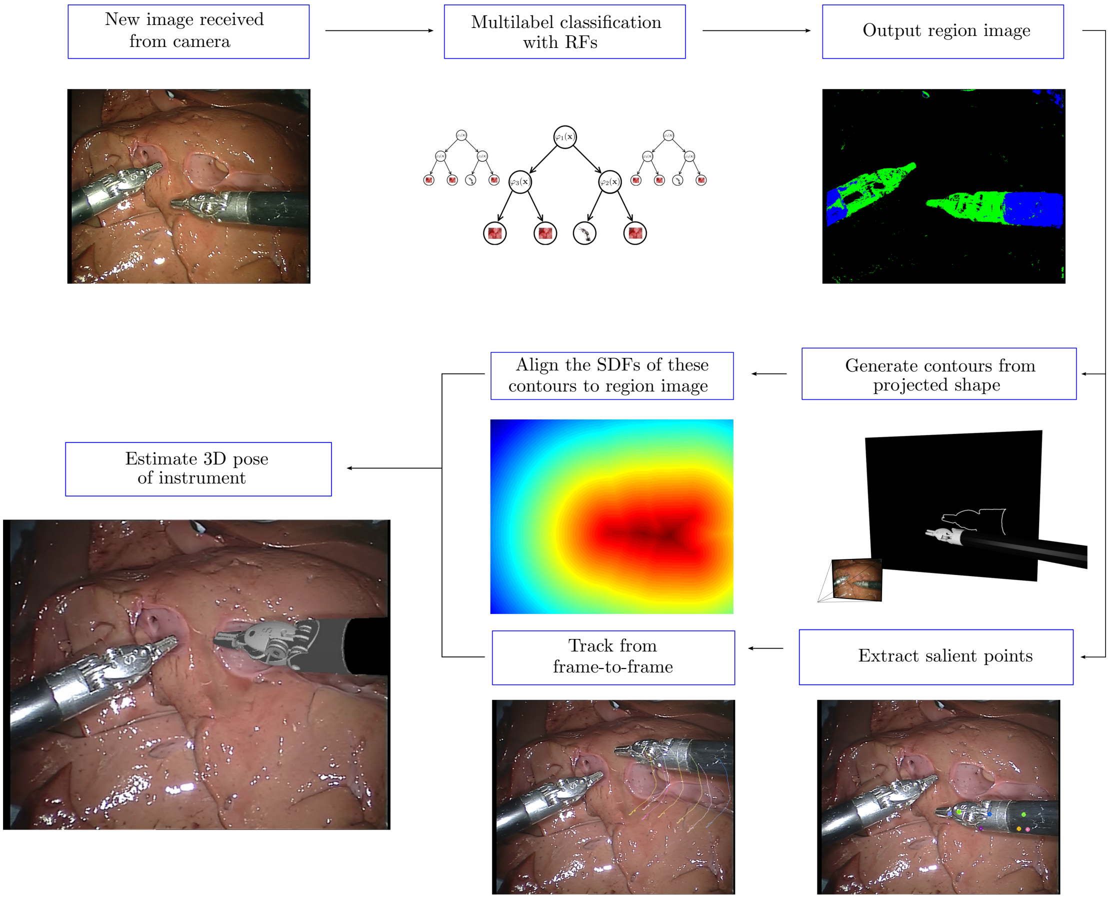


Figure 5.5: An overview of our method. The region based method of Chapter 4 is combined with the feature points to create a tracker where the weaknesses of each method are compensated by the other. The different features are used so that their respective strengths balance the other's weaknesses.

5.4 Experiments

To demonstrate the benefit of using interior features and also to understand which interior feature provide the best performance, we perform quantitative and qualitative analysis between the MR tracker from Chapter 4 using either SIFT or LK features. We perform the analysis on the same datasets as the experiments in Chapter 4 and present the results in the same way using trajectory plots of translation and rotation as well as images with instrument renderings overlaid for visual comparison. Numerical results are presented in Tables 5.1, 5.2 and 5.3 and these also show a comparison to the MR tracker results in the overall table.

5.4.1 Implementation Details

We use the OpenCV implementations of SIFT and pyramidal LK [178]. When computing the SIFT features, we filter the matches by discarding all SIFT matches where the feature distance was more than twice the distance of the best matching feature [98] and we prevent features from being more than 40 pixels apart between frames. This allows us to filter out poor matches and also decreases the matching time, which scales as $O(n^2)$, where n is the number of detected features as we use brute force matching. For the optical flow computations, we use a fixed window size of $(31, 31)$ and allow up to 20 Newton steps for convergence between the patches. We initialized salient optical flow features using Shi-Tomasi features [182], where we allow up to 50 features at initialization and we use a mask generated by the projection of the model to avoid initializing features not on the instrument body. If a feature is tracked off the surface of the instrument or off the edge of the image, which is not a common occurrence, we cease to track it.

5.4.2 Ex-Vivo Experiments

In dataset 1, both methods have similar accuracy however, there are some parts of the sequence after frame 200 when the error in the LK tracker increases compared with the SIFT method. This is likely due to the LK method's superior feature matching incorrectly tracking multiple points on the surface of the tissue as it moves to occlude the instrument tip after this frame. The SIFT method maintains fewer matches and this leads to only one or two matches being found on the head as it begins to move behind the tissue. Dataset 2 shows a much improved performance when using LK features, with some small errors noticeable during the period of occlusion when these features are of less use than the region features. This effect can be seen in the inaccuracies in the roll rotation (r_x) which can be seen in Figure 5.9. Datasets 3 and 4 show greatly improved accuracy when using point features, as the estimated rotation follows the ground truth much more closely, however in dataset 3 the SIFT feature tracker rotation error increases greatly after frame 400 where Figure 5.12 shows the misalignment. This occurs because very few feature points are tracked on the tip of the instrument and the method cannot prevent r_y rotation error when the shaft goes out of view. In normal frames, even without point features, the region cues would prevent this misalignment but they are reduced when the shaft is not visible. Datasets 5, 6 and 7 show improvement when using LK features which prevents large r_x errors compared with the SIFT features.

Dataset	$t_x(mm)$	$t_y(mm)$	$t_z(mm)$	$r_x(rads)$	$r_y(rads)$	$r_z(rads)$
Dataset 1 - MR LK	1.00 ± 0.86	1.57 ± 1.18	8.38 ± 6.02	0.50 ± 0.54	0.20 ± 0.11	0.36 ± 0.34
Dataset 2 i - MR LK	0.86 ± 0.77	0.93 ± 0.78	3.88 ± 3.29	0.11 ± 0.11	0.05 ± 0.03	0.03 ± 0.02
Dataset 2 ii - MR LK	1.42 ± 1.29	2.43 ± 1.28	5.11 ± 4.24	0.93 ± 0.54	0.12 ± 0.09	0.20 ± 0.10
Dataset 3 - MR LK	0.85 ± 0.53	0.67 ± 0.56	3.38 ± 2.20	0.18 ± 0.11	0.05 ± 0.03	0.10 ± 0.05
Dataset 4 - MR LK	0.72 ± 0.56	0.69 ± 0.60	3.83 ± 3.59	0.09 ± 0.08	0.05 ± 0.05	0.05 ± 0.06
Dataset 5 - MR LK	1.18 ± 0.79	1.88 ± 1.38	13.08 ± 8.15	0.40 ± 0.52	0.18 ± 0.21	0.33 ± 0.54
Dataset 6 i - MR LK	0.68 ± 0.53	0.85 ± 0.77	4.24 ± 2.62	0.37 ± 0.17	0.07 ± 0.06	0.05 ± 0.07
Dataset 6 ii - MR LK	0.45 ± 0.08	0.06 ± 0.05	4.20 ± 0.31	0.29 ± 0.02	0.05 ± 0.01	0.07 ± 0.02
Dataset 7 i - MR LK	0.44 ± 0.35	0.63 ± 0.54	4.55 ± 3.17	0.13 ± 0.08	0.08 ± 0.05	0.01 ± 0.01
Dataset 7 ii - MR LK	0.64 ± 0.41	0.49 ± 0.51	3.62 ± 2.78	0.04 ± 0.04	0.02 ± 0.01	0.07 ± 0.05

Table 5.1: Errors for 3D pose estimation for multi-region (MR) level set trackers when using LK optical flow features. The translation and rotation errors for each dataset are shown where datasets with two instruments are shown separately as Dataset n i or Dataset n ii for the left and right instrument respectively. The values show are the mean error over all frames ± the standard deviation.

Dataset	$t_x(mm)$	$t_y(mm)$	$t_z(mm)$	$r_x(rads)$	$r_y(rads)$	$r_z(rads)$
Dataset 1 - MR S	2.69 ± 1.92	3.08 ± 2.06	5.14 ± 3.11	0.28 ± 0.16	0.28 ± 0.07	0.34 ± 0.12
Dataset 2 i - MR S	2.79 ± 2.89	1.49 ± 1.51	11.39 ± 7.66	0.64 ± 0.59	0.21 ± 0.17	0.17 ± 0.11
Dataset 2 ii - MR S	1.53 ± 0.75	2.44 ± 1.21	4.65 ± 3.65	0.79 ± 0.60	0.16 ± 0.06	0.24 ± 0.06
Dataset 3 - MR S	1.27 ± 1.07	1.15 ± 0.91	4.13 ± 2.87	0.84 ± 0.51	0.07 ± 0.07	0.07 ± 0.07
Dataset 4 - MR S	0.57 ± 0.46	0.73 ± 0.62	2.02 ± 1.42	0.57 ± 0.33	0.05 ± 0.03	0.07 ± 0.06
Dataset 5 - MR S	3.26 ± 1.56	2.75 ± 1.74	12.22 ± 4.79	0.27 ± 0.19	0.27 ± 0.09	0.20 ± 0.05
Dataset 6 i - MR S	1.14 ± 0.62	0.85 ± 0.78	5.71 ± 4.49	0.62 ± 0.36	0.07 ± 0.04	0.05 ± 0.05
Dataset 6 ii - MR S	0.75 ± 0.51	0.58 ± 0.28	5.37 ± 2.73	0.52 ± 0.31	0.17 ± 0.08	0.20 ± 0.09
Dataset 7 i - MR S	0.91 ± 0.42	0.39 ± 0.29	2.58 ± 1.64	0.32 ± 0.23	0.17 ± 0.04	0.01 ± 0.01
Dataset 7 ii - MR S	0.78 ± 0.64	0.48 ± 0.49	2.25 ± 1.53	0.20 ± 0.14	0.07 ± 0.03	0.17 ± 0.13

Table 5.2: Errors for 3D pose estimation for multi-region (MR) level set trackers when using SIFT features. The terms in this table are the same as table 5.1

	$t_x(mm)$	$t_y(mm)$	$t_z(mm)$	$r_x(rads)$	$r_y(rads)$	$r_z(rads)$
Mean error MR LK	**0.82 ± 0.62**	**1.02 ± 0.76**	**5.43 ± 3.64**	**0.30 ± 0.22**	**0.09 ± 0.06**	0.13 ± 0.13
Mean error MR S	1.57 ± 1.08	1.39 ± 0.99	5.55 ± 3.39	0.50 ± 0.34	0.15 ± 0.07	0.15 ± 0.07
Mean error MR	1.80 ± 1.48	1.55 ± 1.07	6.14 ± 3.60	1.10 ± 0.77	0.14 ± 0.07	**0.13 ± 0.06**

Table 5.3: Overall errors of 3D pose estimation for multiple regions (MR) level set tracking with no point features, with SIFT (S) or with LK optical flow over all datasets. The values shown are the mean error over all frames ± the standard deviation. The bold values show the method with the lowest average error for the given DOF.

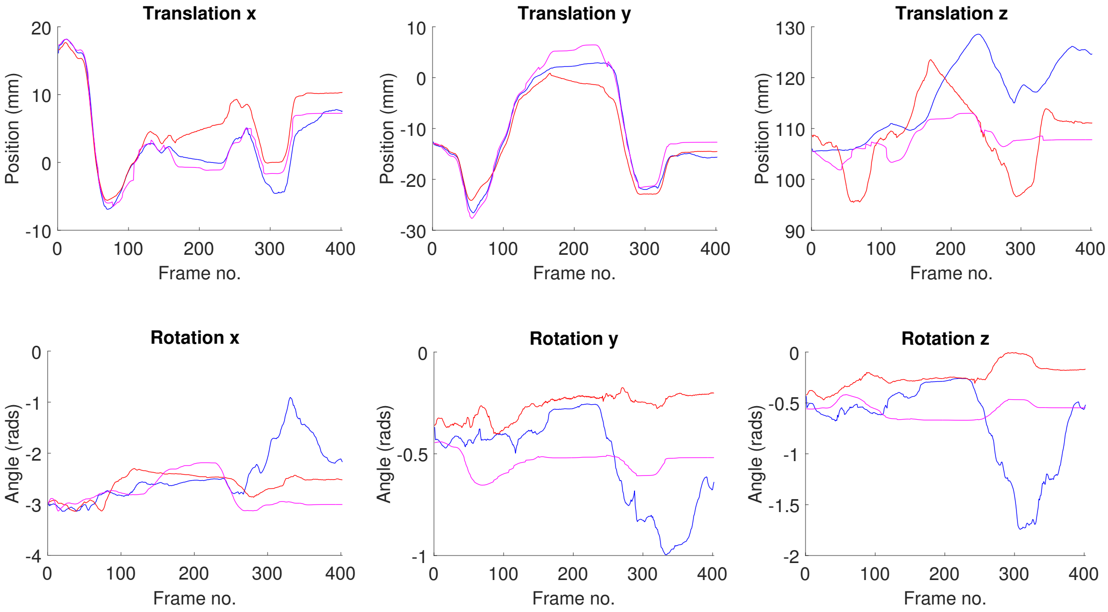


Figure 5.6: MR SIFT Tracker, MR LK Tracker, Ground Truth. The analysis of the translation and rotation of of the instrument for ex-vivo Dataset 1. The accuracy decreases between frames 150 and 250 as this period is when the instrument is occluded partially by the tissue.

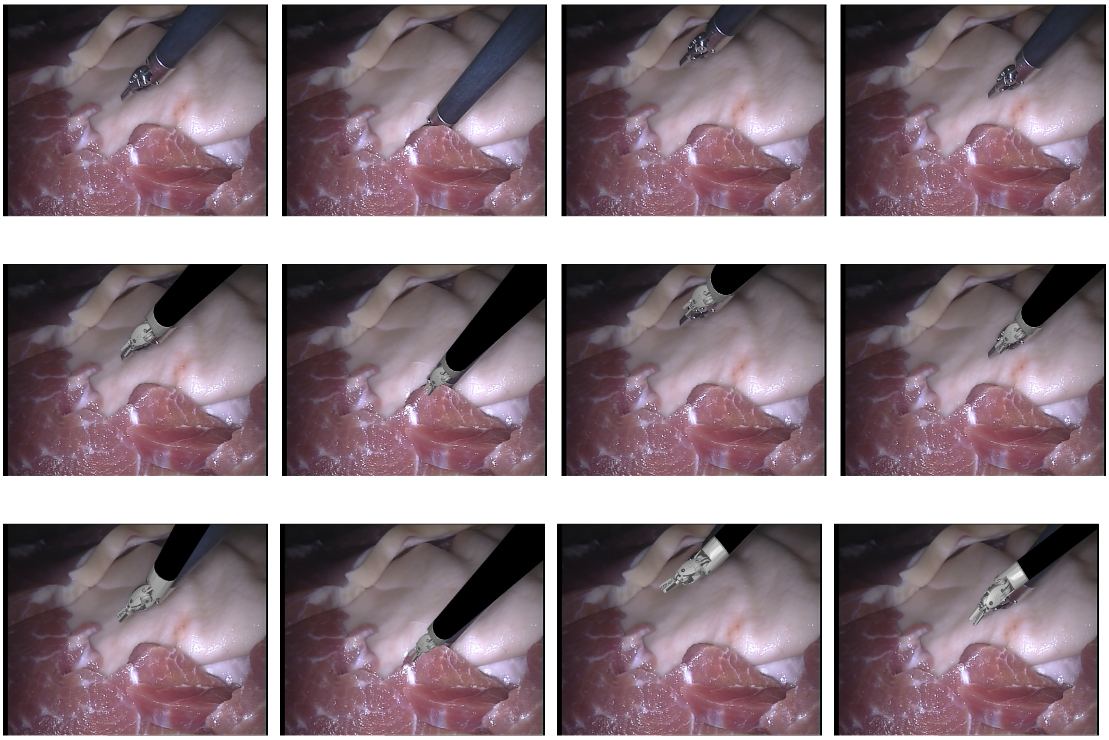


Figure 5.7: Sampled frames 100, 200, 300 and 350 from dataset 1 are shown in the top row and the corresponding frames from the MR SIFT tracker in row 2 and from the MR LK tracker in row 3.

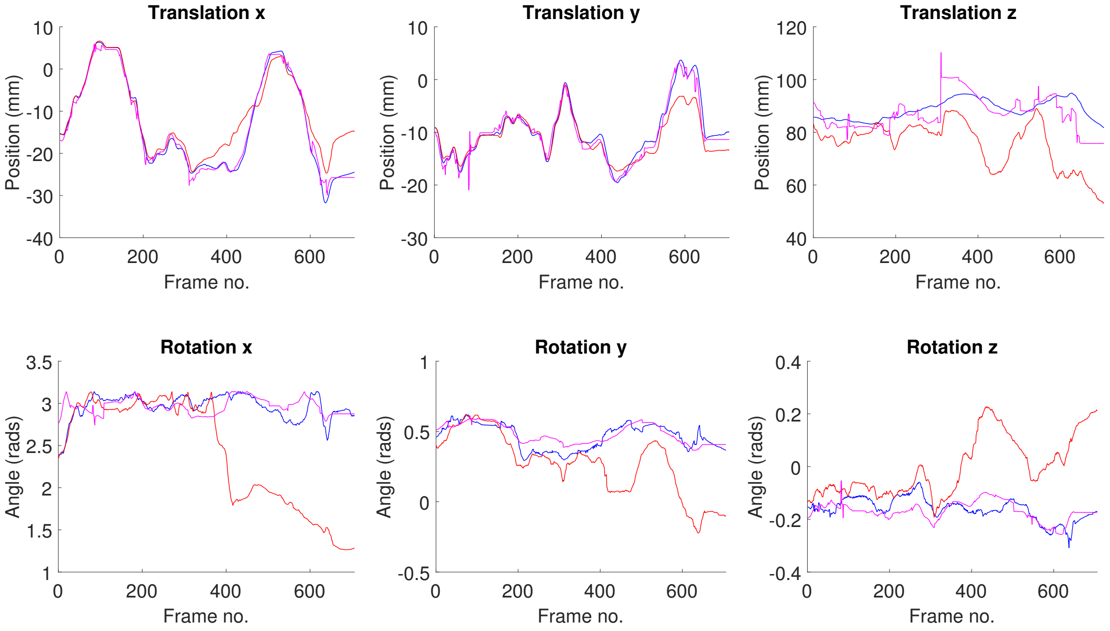


Figure 5.8: MR SIFT Tracker, MR LK Tracker, Ground Truth. Trajectory plots for the left instrument for ex-vivo dataset 2 using multiple regions level set trackers with either SIFT or LK tracking. Accuracy over this sequence is generally quite good, as the left instrument is not affected by the occlusion.

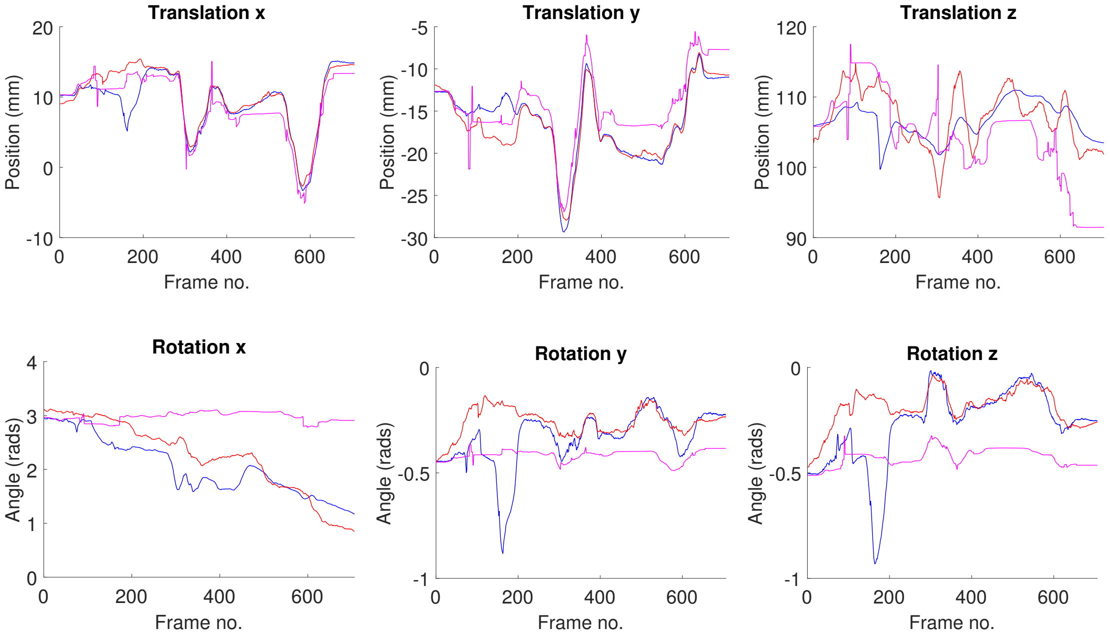


Figure 5.9: MR SIFT Tracker, MR LK Tracker, Ground Truth. Trajectory plots for the the right instrument for ex-vivo dataset 2. There is significant error around frame 180 as the instrument is fully occlude by the left instrument at this point. However, tracking after this point quickly recovers.

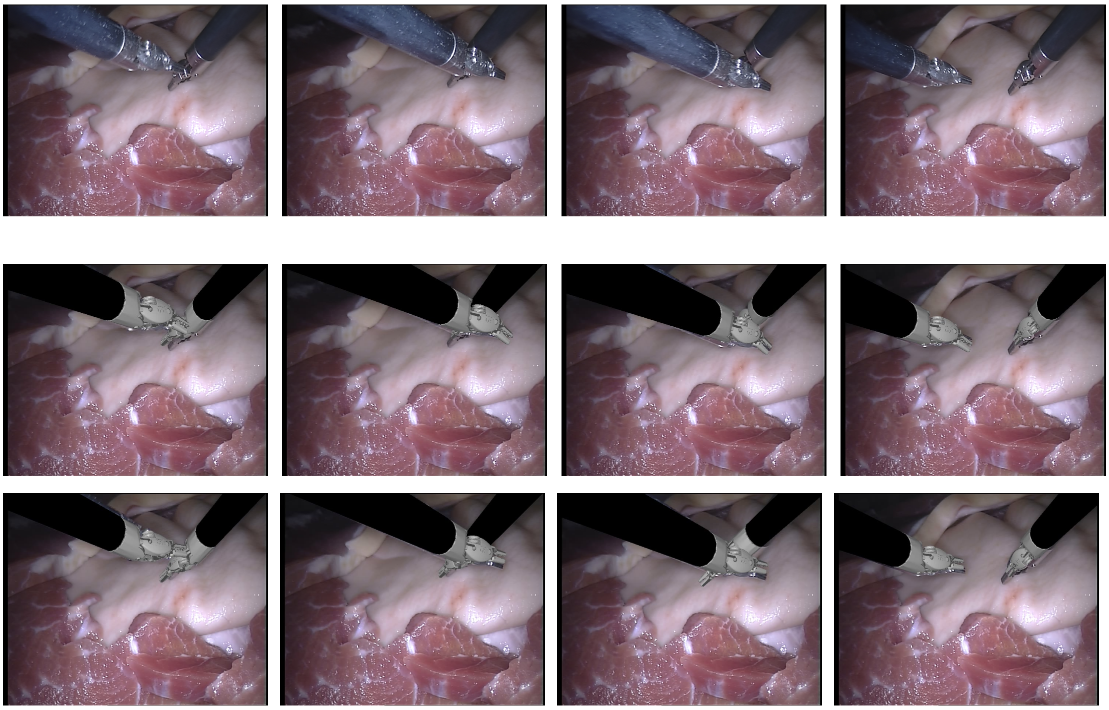


Figure 5.10: Qualitative analysis from ex-vivo dataset 2 showing the original frames 50, 100, 150, 250 in row 1, the corresponding frames from the MR SIFT tracker in row 2 and from the MR LK tracker in row 3. In frame 100 there is improvement in using the interior features as the roll rotation of the left instrument is estimated more accurately.

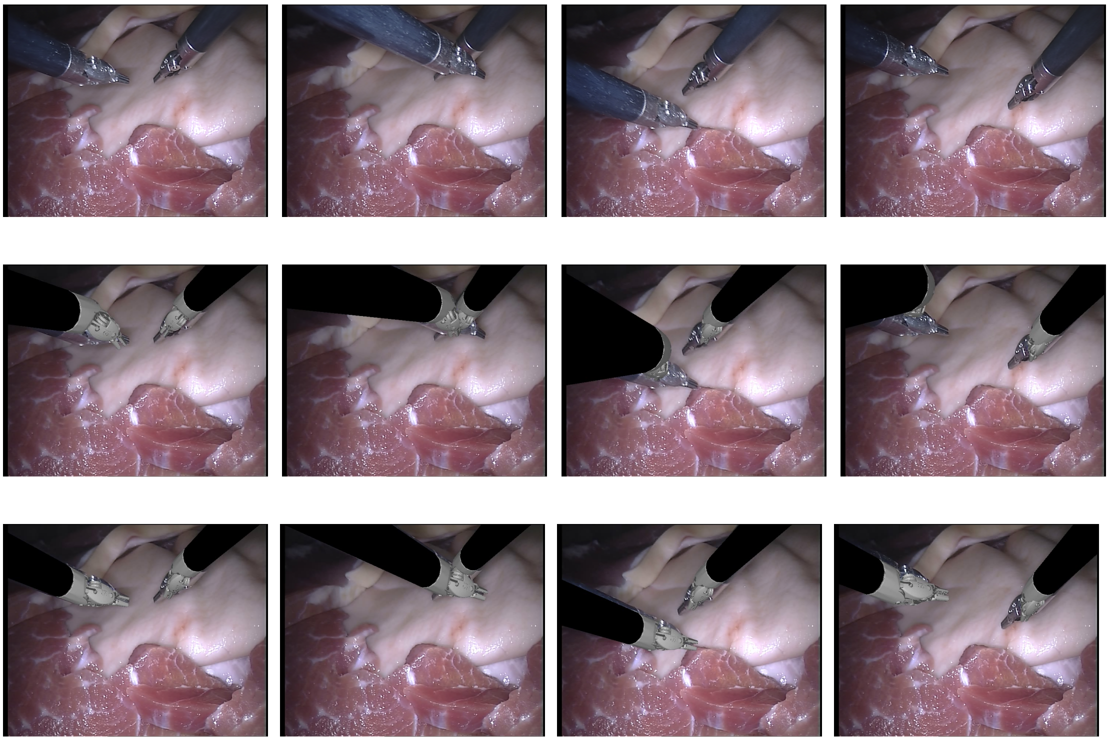


Figure 5.11: Qualitative analysis from ex-vivo dataset 2 showing the original frames 400, 500, 600 and 700 in row 1, the corresponding frames from the MR SIFT tracker in row 2 and from the MR LK tracker in row 3. The SIFT method begins to show tracking failure at frames 600-700 which occurs mostly due to accumulation of small errors over the whole sequence leading to a large misalignment when the second instrument occlusion occurs.

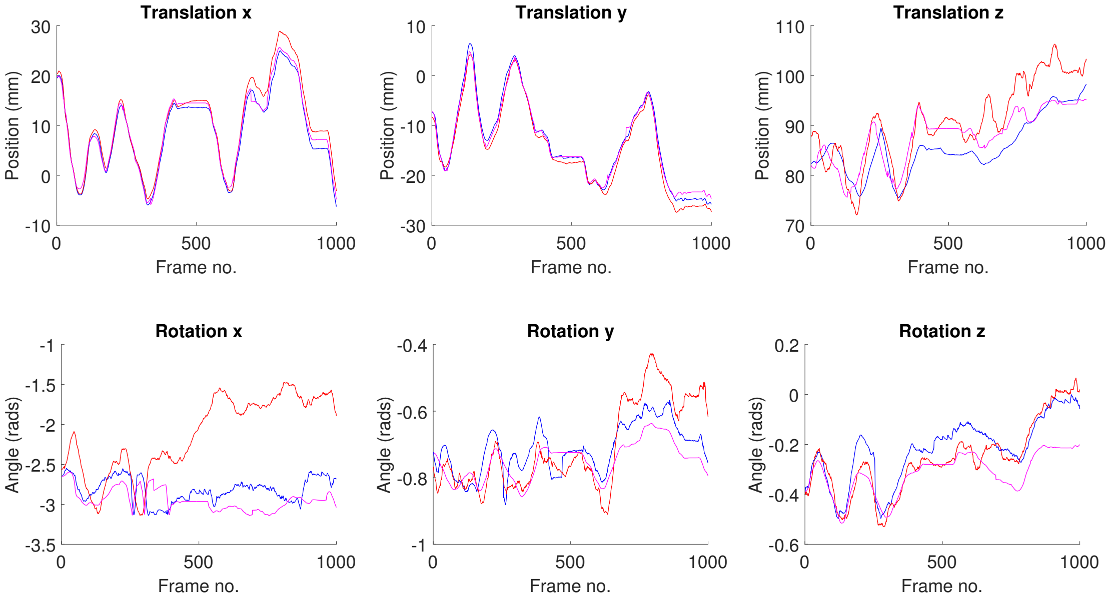


Figure 5.12: SR Tracker, MR Tracker, Ground Truth. The quantitative analysis of the translation and rotation of ex-vivo dataset 3. The accuracy is quite high in this dataset, particularly in the t_z and r_y and r_z directions. There is some r_x errors in the SIFT tracker after frame 500. This is likely caused by drift errors accumulated from earlier frames.

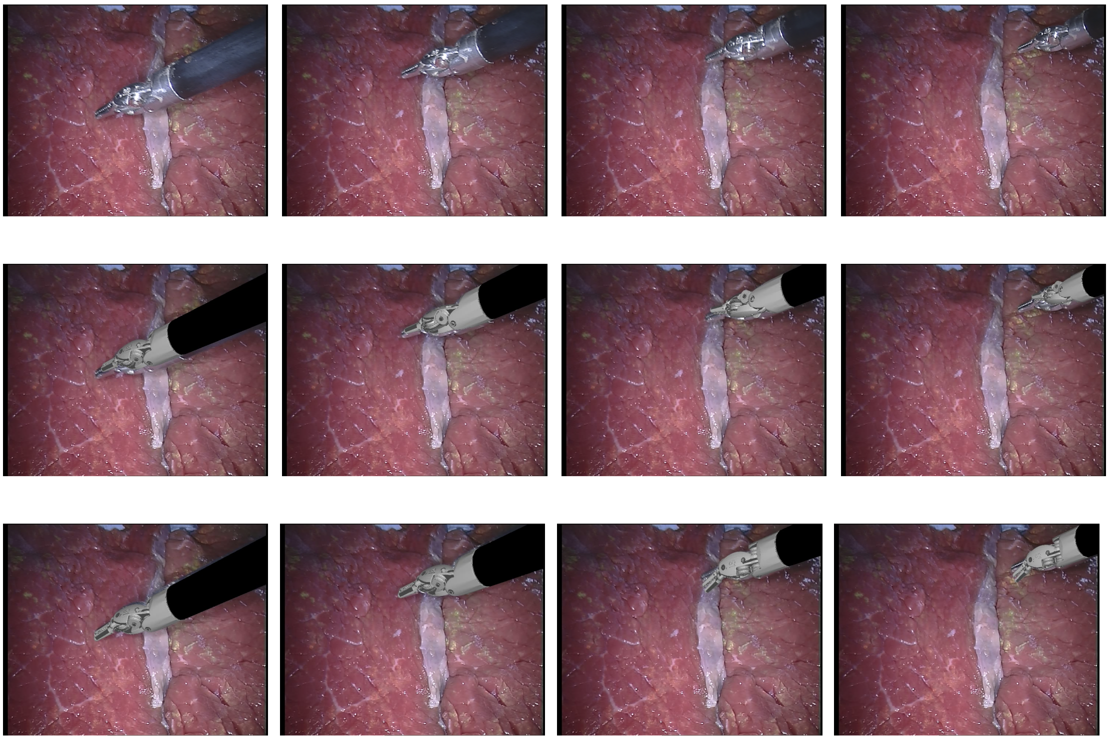


Figure 5.13: Sampled frames from ex-vivo dataset 3 showing 100, 200, 550 and 850 in the top row and the corresponding frames from the MR SIFT tracker in row 2 and from the MR LK tracker in row 3. The r_x rotation error for the SIFT tracker is clearly visible in frames 550 and 850.

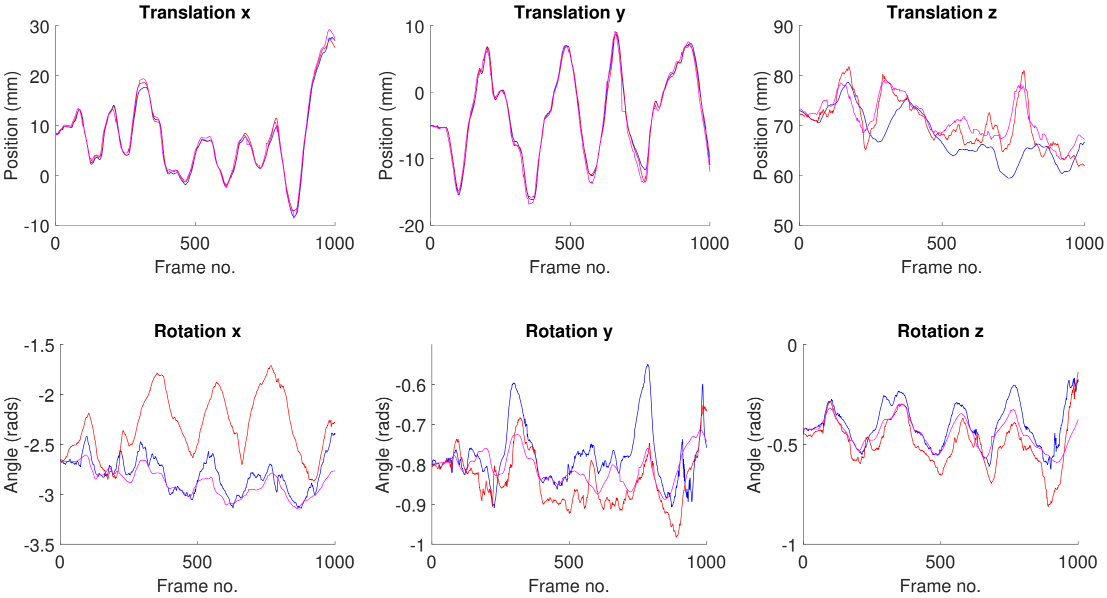


Figure 5.14: SR Tracker, MR Tracker, Ground Truth. The analysis of the translation and rotation of ex-vivo dataset 4. Much like in the region based tracker in the previous chapter, this dataset records very high accuracy but the interior feature trackers have much better r_x rotational accuracy.

Figure 5.15: Sampled frames from dataset 4 showing 100, 200, 550 and 850 in the top row and the corresponding frames from the MR SIFT tracker in row 2 and from the MR LK tracker in row 3. In frame 850 the r_x rotation is observed to still match closely with the visual data.

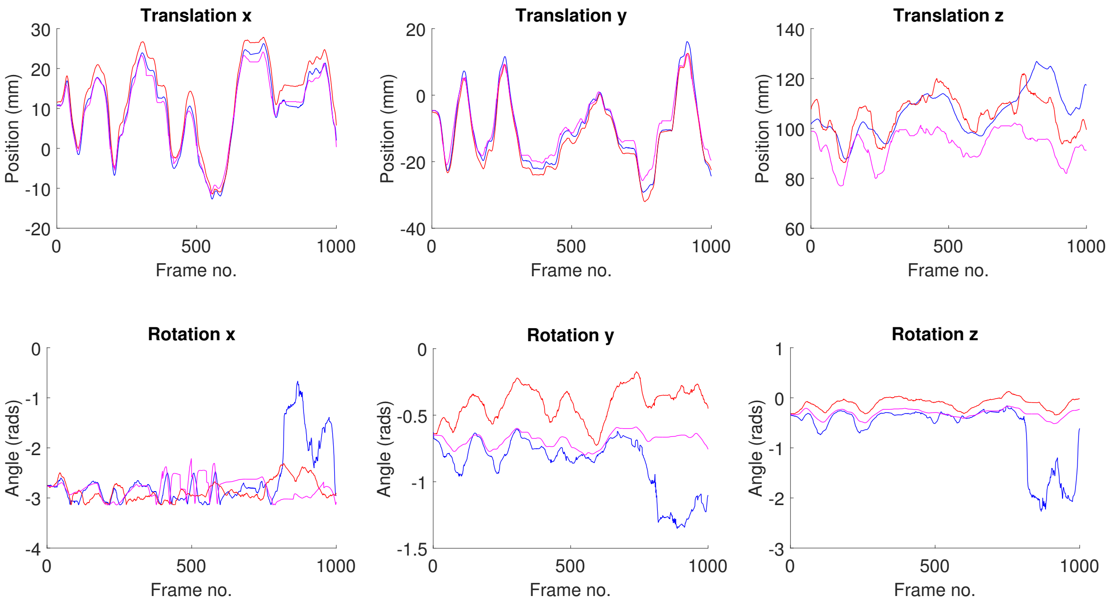


Figure 5.16: MR SIFT Tracker, MR LK Tracker, Ground Truth. The analysis of the translation and rotation of ex-vivo dataset 5. The error is mostly quite low over this dataset but some error occurs after frame 850 as the instrument shaft moves out of view.

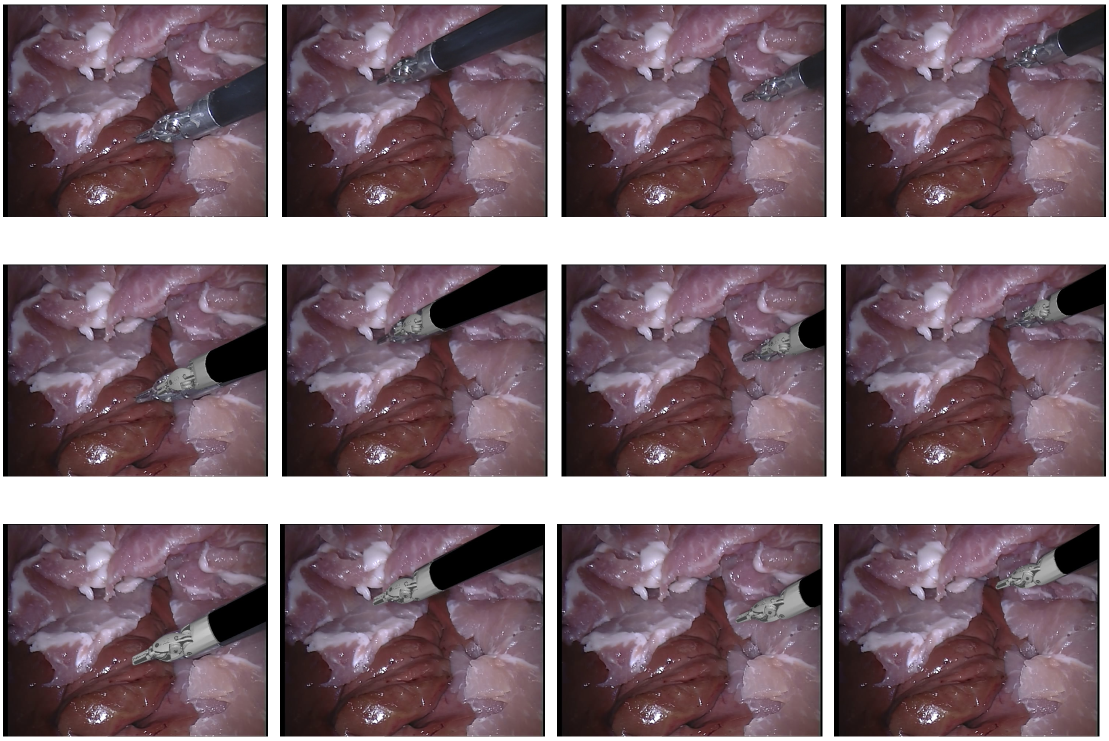


Figure 5.17: Sampled frames 100, 200, 550 and 850 (top row) and the corresponding frames from the MR SIFT tracker in row 2 and from the MR LK tracker in row 3. Although the color classification around the tip is not strong in this dataset, which created higher errors when using the region only tracker, the surface features help to reduce this error.

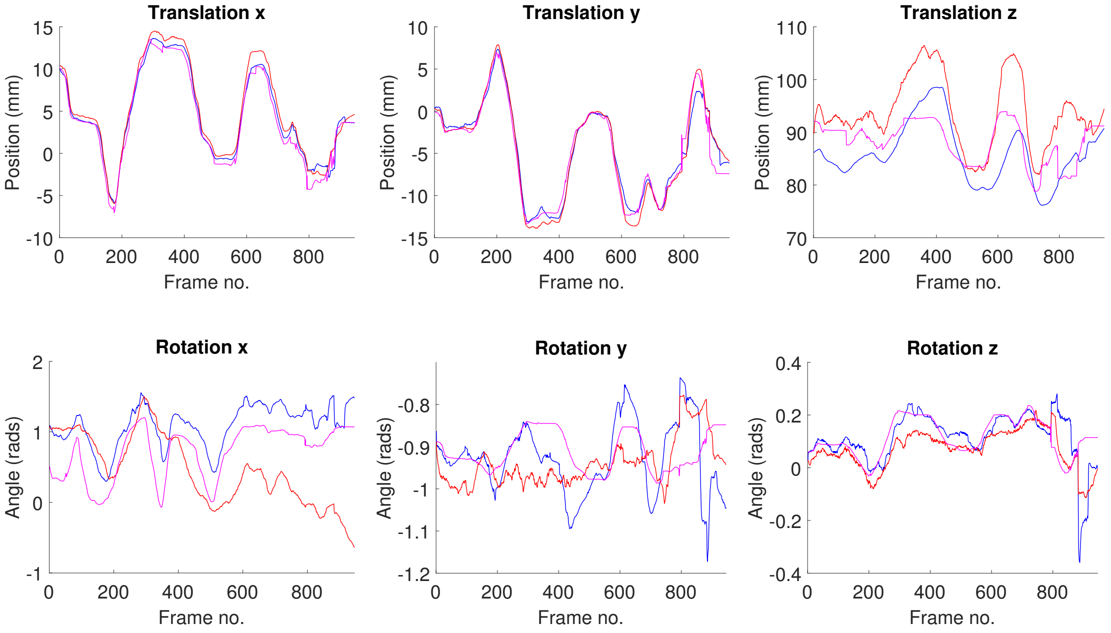


Figure 5.18: MR SIFT Tracker, MR LK Tracker, Ground Truth. Quantitative analysis of the right instrument for ex-vivo dataset 6. There are some errors in r_y and r_z which occur when the instruments slightly occlude one another.

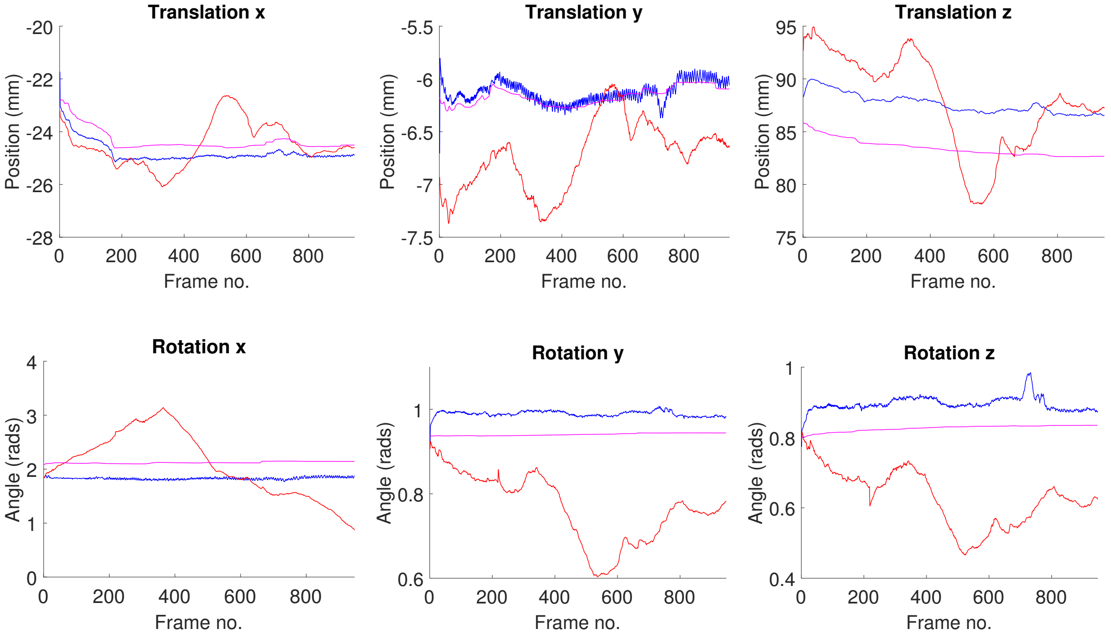


Figure 5.19: MR SIFT Tracker, MR LK Tracker, Ground Truth. The trajectories for the left instrument for ex-vivo dataset 6. The instrument is tracked quite accurately by both methods where the apparent larger rotational errors occur due to the range of the axis which covers only 0.2-0.3 radians due to the limited rotational movements in this dataset.

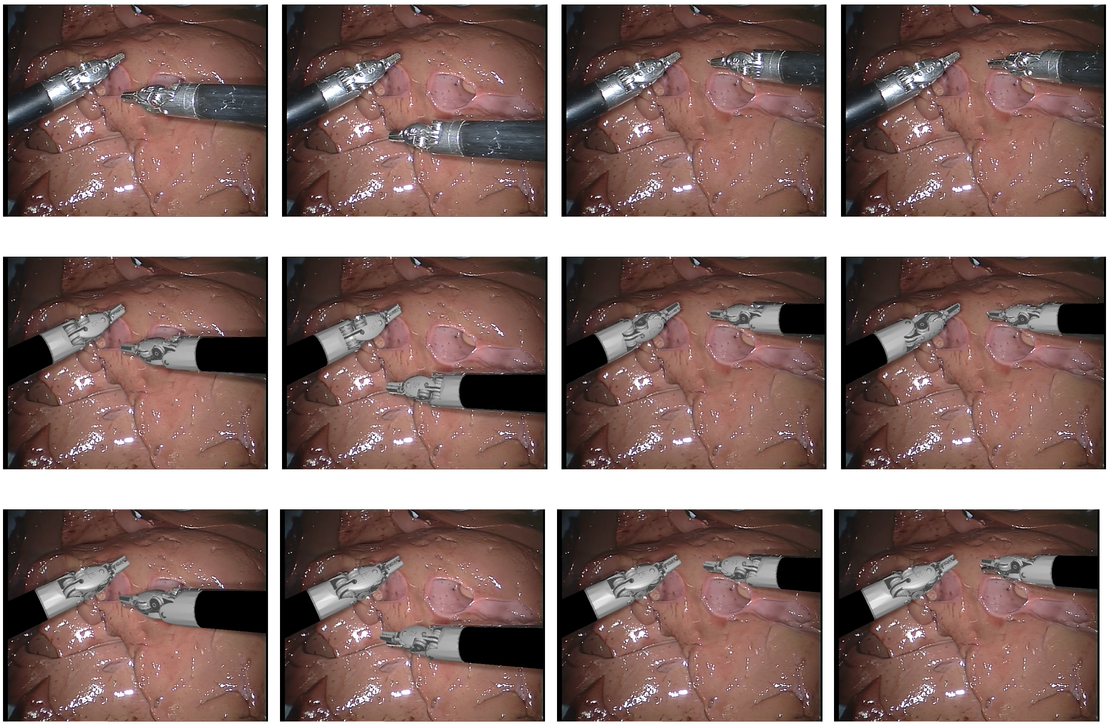


Figure 5.20: Qualitative analysis from ex-vivo dataset 6 showing frames 100, 200, 350 and 400. Row 1 shows the original frames, row 2 shows the MR SIFT tracker and row 3 shows the MR LK tracker. The results for frame 400 show that as the instrument rotates around the x axis, the tracking is maintained. This contrasts to the region only trackers in the previous chapter.

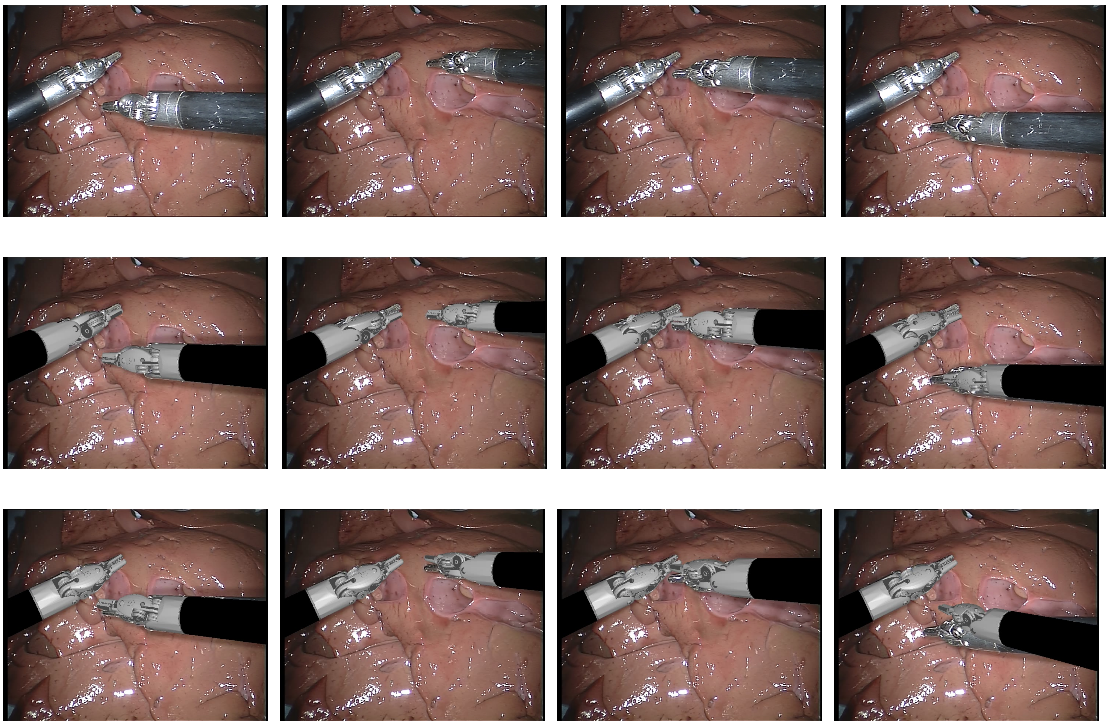


Figure 5.21: Qualitative analysis from ex-vivo dataset 6 showing frames 500, 650, 750, and 850. Row 1 shows the original frames, row 2 shows the MR SIFT tracker and row 3 shows the MR LK tracker. There is some misalignment of the right instrument in frame 850 where some error was introduced as the instruments passed close together.

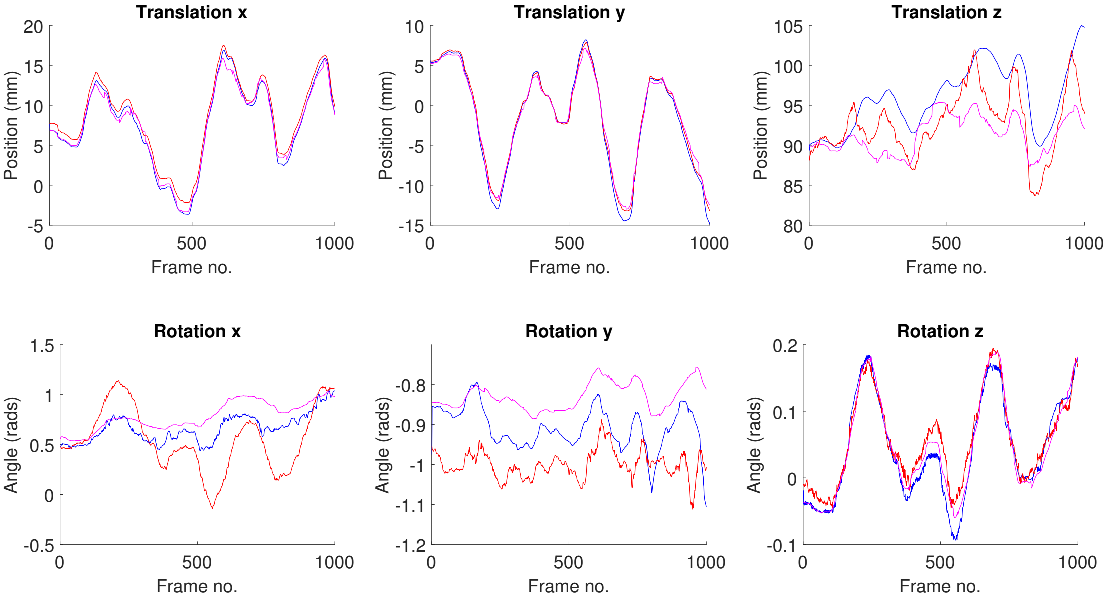


Figure 5.22: MR SIFT Tracker, MR LK Tracker, Ground Truth. Quantitative analysis of the right instrument for ex-vivo dataset 7 for the right instrument. Both trackers have good translational accuracy but there are some smaller rotational errors in r_z.

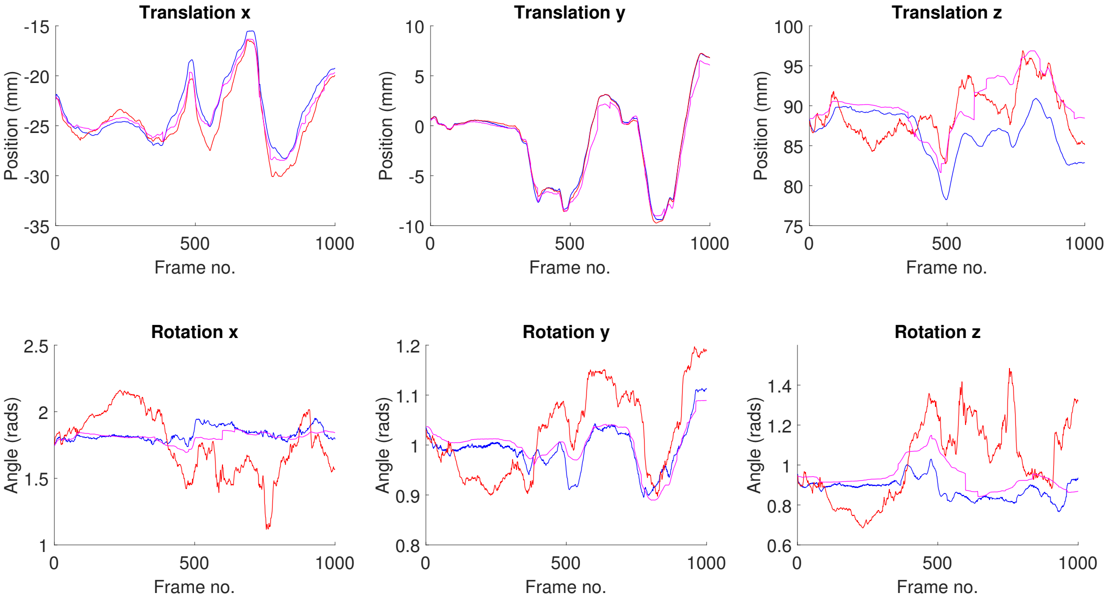


Figure 5.23: MR SIFT Tracker, MR LK Tracker, Ground Truth. The analysis of the translation and rotation of the left instrument for ex-vivo dataset 7. Both trackers have good accuracy over this dataset, which is confirmed by the visual results in Figure 5.24 and 5.25

.

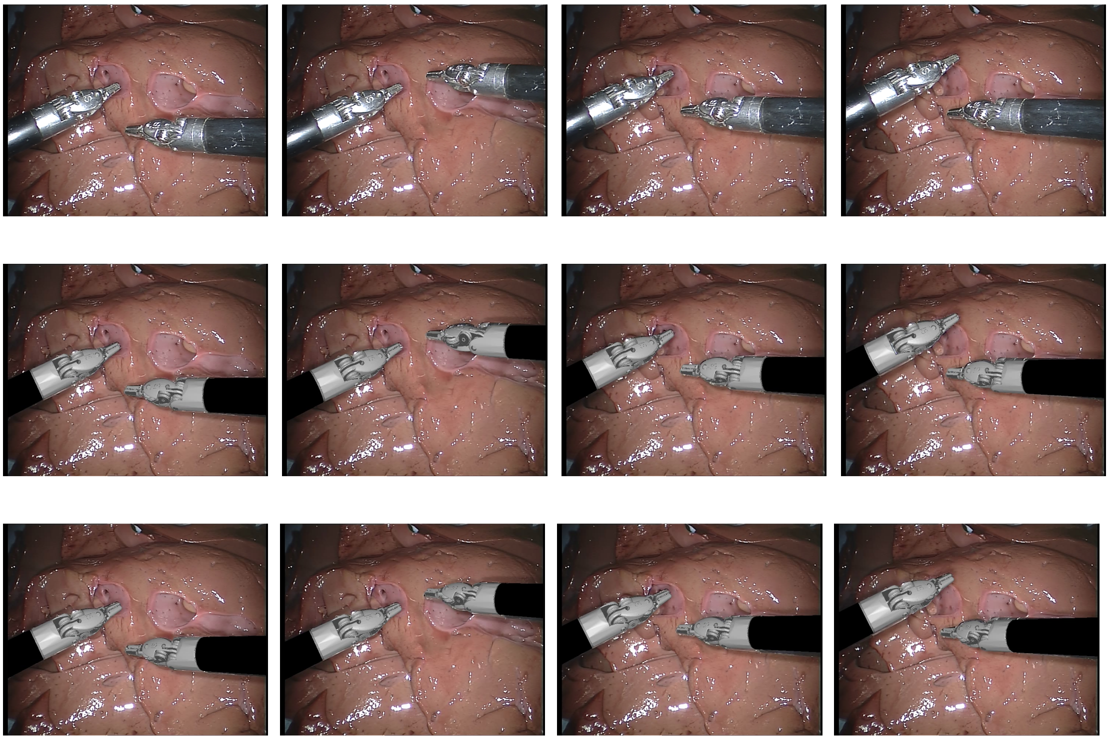


Figure 5.24: Qualitative analysis from ex-vivo dataset 7 showing frames 100, 200, 350 and 400. Row 1 shows the original frames, row 2 shows the MR SIFT tracker and row 3 shows the MR LK tracker. The instrument motion is quite simple and the color classification is good resulting in very accurate alignment.

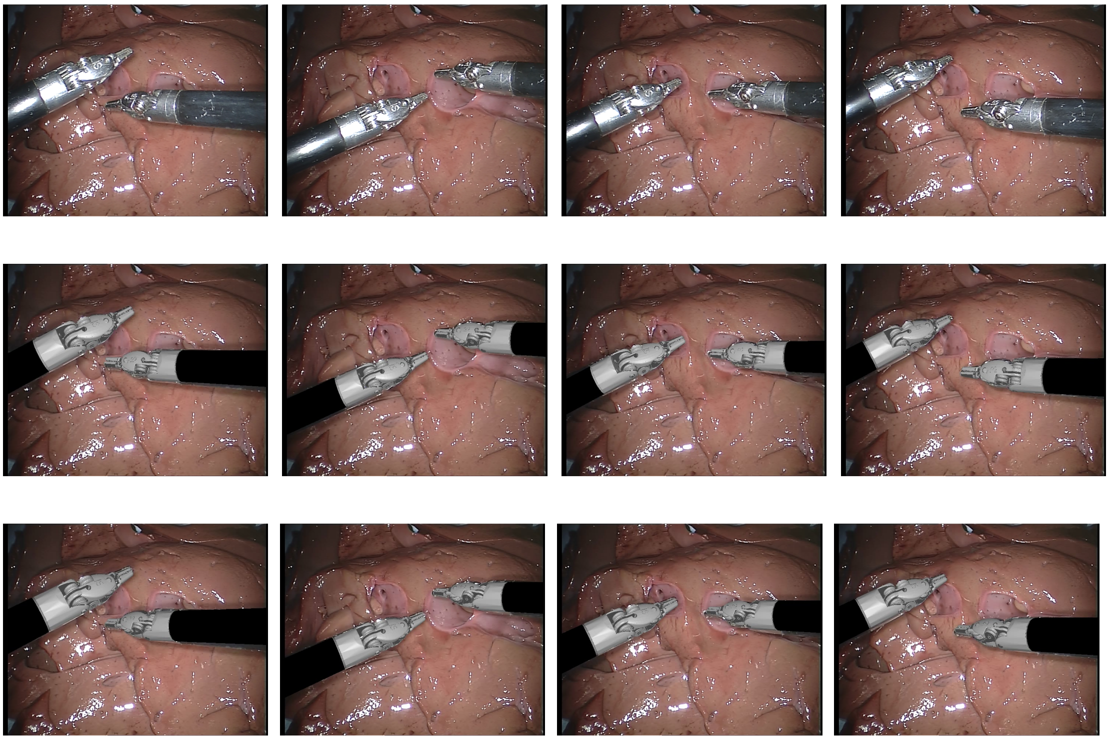


Figure 5.25: Qualitative analysis from ex-vivo dataset 7 showing frames 500, 650, 750, and 850. Row 1 shows the original frames, row 2 shows the MR SIFT tracker and row 3 shows the MR LK tracker. All frames show good visual alignment, even during periods when the instrument undergoes roll rotation.

5.4.3 In-Vivo Experiments

We also perform qualitative evaluation on in-vivo data during a robot-assisted prostatectomy. This evaluation shows that the proposed MR LK tracker is able to track robotic instruments through realistic sequences as the estimated pose clearly lines up well with the images captured by the camera.

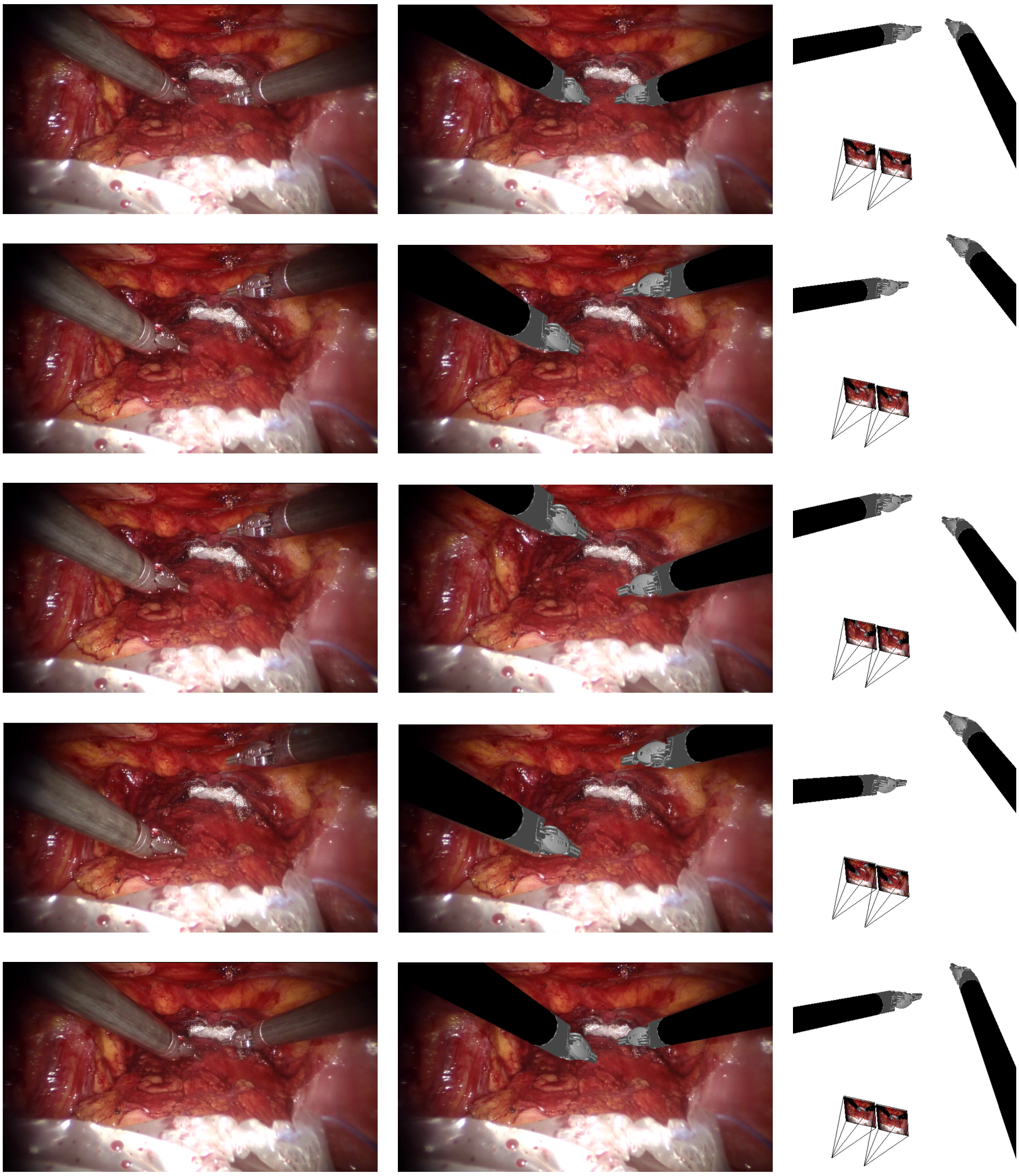


Figure 5.26: In each row we show the original frame in the left column, the frame with instrument rendering in the center column and 3D rendering in the right column. Visual inspection shows that the instruments align well with the visual data.

5.4.4 Camera Tracking

An interesting application of camera tracking arises when attempting to track robotic surgical instruments in 3D due to the forced asynchronous motion between the camera and instruments on systems such as da Vinci. This allows the cases of relative motion to be distinguished as either a moving camera and static instruments or a static camera and moving instruments. At the start of camera motion, which can be determined using the head sensor on a da Vinci or potentially optically using scene flow, the vertices of the instrument models X_m are transformed from model coordinates into a single world coordinate system $\mathcal{F}_w$ which can be defined at the origin of the camera coordinate system $\mathcal{F}_{cam}$. We then define an energy function over all observable instruments leading to the following energy function:

$$E(\boldsymbol{\theta}_c) = \sum_{i \in \mathcal{I}} E_i(\boldsymbol{\theta}_c) \tag{5.8}$$

where each i is a single instrument in the set of all observable instruments $\mathcal{I}$ and E_i is the energy defined in Equation 5.4 except the terms are parameterized by the camera pose with respect to the $\mathcal{F}_w$. To perform the tracking we use the MR LK tracking system developed in this chapter.

We demonstrate the accuracy of this method with an ex-vivo sequence where 2 Large Needle Driver instruments maintain a static position and the surgical camera is moved around the scene. The same kinematic capture and calibration correction as in section 4.6.2 is used to acquire the pose of the camera and all updates are assuming that the camera pose relative to the robot coordinate system is known in the first frame.

Dataset	$t_x(mm)$	$t_y(mm)$	$t_z(mm)$	$r_x(rads)$	$r_y(rads)$	$r_z(rads)$
Dataset 1 - MR LK	1.64 ± 1.40	0.67 ± 0.55	3.28 ± 2.38	0.01 ± 0.01	0.01 ± 0.01	0.01 ± 0.01

Table 5.4: The numerical accuracy of the camera tracking when using the MR LK tracking system. The translation and rotation errors for the dataset are shown where the values indicate the mean error over all frames ± the standard deviation.

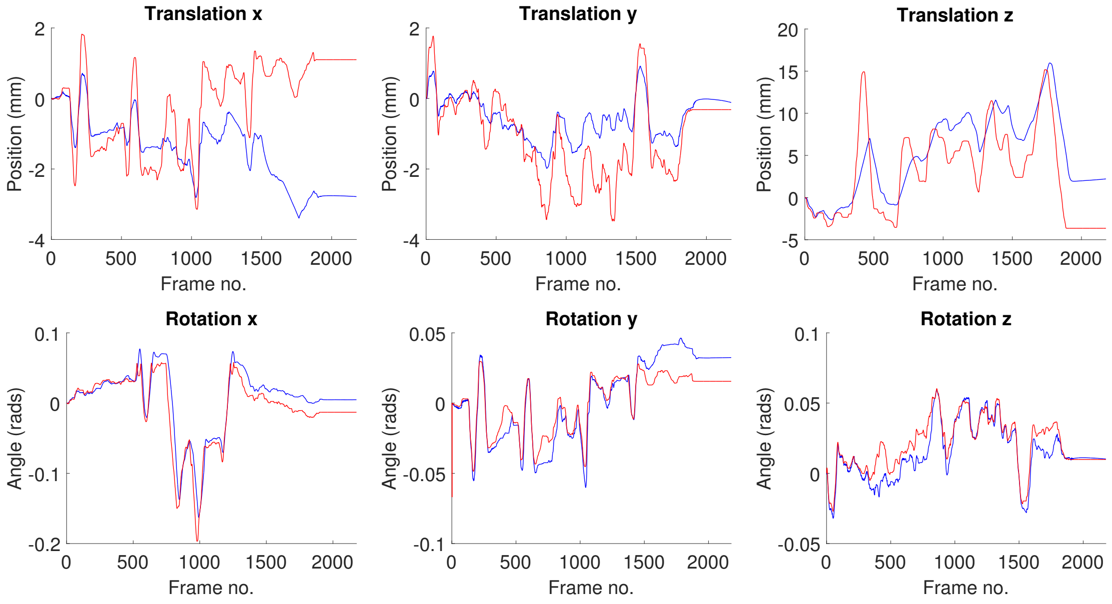


Figure 5.27: Ground Truth, MR LK Tracker. The analysis of the translation and rotation of the camera shows that there is some t_x and t_y error, which is unusual compared with the instrument trackers.

Figure 5.28: Sampled frames 100, 600, 850, 1000, 1200, 1400, 1600 and 1850 from the ex-vivo camera tracking dataset are shown in row 1. Row 2 shows the MR LK level set tracker. We render a coordinate system in the field of view which is updated as the pose of the camera changes and this remains static relative to the anatomy.

5.5 Conclusion

In this chapter we have demonstrated a method for solving some of the challenges we faced in Chapter 4 around tracking the r_x rotation of the instruments and also dealing with situations where the silhouettes were unreliable due to noise in the classification. Through our extensive ex-vivo experiments we have demonstrated that we can reliably track this DOF and show considerable performance increase over using only region based features in Table 5.3. When comparing either SIFT or optical flow features, we find that in nearly every dataset the optical flow tracking outperforms the SIFT features, which is normally due to a much higher number of matched features between frames. However, in dataset 1 there is a noticeable drop in performance as the LK tracker incorrectly tracks some features onto the tissue surface. This is a particular disadvantage of using this type of interior feature as it is not discriminative about the appearance of the patches it tracks. Although this issue does not cause complete tracking failure as the region features and points correctly tracked on the instrument surface successfully track the instrument as it moves out from behind the tissue, it does disrupt the accuracy leaving a large offset in the final frames.

We have also illustrated a novel camera tracking application of our method for robotic surgery, where we demonstrate on a simple ex-vivo setup that it is possible to accurately predict the pose of the surgical camera solely by tracking the instruments. Although a full camera relocalization system would likely make use of all of the background information in estimating pose, using the instruments is a potentially useful augmentation to these systems, which typically ignore the instruments altogether [183]. The instruments themselves would provide useful information when the background undergoes motion or appearance change due to patient breathing or bleeding. The instruments would mostly be immune to these visual challenges and could for instance provide stable tracking while the background model is reinitialized.

A major limitation with the type of feature introduced in this chapter is that they only track relative motion between frames and as such is highly susceptible to drift. This is particularly an issue for the roll rotation of the instruments where if the motion is particularly fast and difficult to track, it is often not possible to recover this DOF using the point features. In principle, the region based method could provide some information to recover this DOF but this has been shown to be unreliable. A second limitation with the work presented in this chapter is that the instrument model is fully rigid, which greatly impacts the application of the method to artificial setups where the instruments are kept rigid. In Chapter 6 we will extend the framework presented in this chapter to tracking fully articulated robotic instruments by using the known kinematic structure of the instruments to guide the optimization.

Chapter 6

Articulated 3D Pose Estimation

6.1 Introduction

Extending visual tracking methods to objects which do not have a static shape has become an exciting area of research [184, 143]. Generative scene and object modelling has heavily relied on assumptions of static target shape [172, 185] which allowed correspondences between the model and the data to be more easily found. Discriminative methods showed good performance on human body and hand datasets [117, 186] but required large quantities of training data to reach acceptable performance which limits their application in MIS where training sets are challenging to acquire.

Tracking an articulated or deforming object with a generative model is usually achieved by selecting a base coordinate system for the model and concurrently to estimating the rigid camera to model transform ${}^{cam}\mathbf{T}_{model}$, estimating a separate transform ${}^{model}\mathbf{T}_{warp}$ which deforms the model vertices relative to the base. For many generic or highly deformable objects, it is desirable to describe this transform in a way that allows significant flexibility in how the model can transform. This can be achieved by modelling the entire deformation as an interpolated series of rigid body transforms which are estimated online [184] or alternatively by learning a low dimensional deformation space from a training set [143]. For MIS, manufactured robotic manipulators such as surgical instruments have a known set of possible transformations which constrain the vertices of each joint to rotate or translate around or along a single axis (see Figure 6.1). Hence, this allows the warping transform to be represented as a composition of several single axis transforms ${}^{n-1}\mathbf{T}_n$ which are applied consecutively to different subsets of the model vertices.

The rigid 3D instrument model used in Chapters 4 and 5 is restrictive for tracking in both robotic and laparoscopic surgery. In RMIS, systems such as da Vinci have multiple DOFs (see Figure 6.1a) which enable the instrument to couple closely to human wrist motion which aids with complex surgical tasks such as stitching which require highly dexterous movements. Laparoscopic instruments are much simpler and have a single axis clasper for grasping or cutting tissue (see Figure 6.1b). Modelling the instrument as a rigid body fails to account for these degrees of freedom meaning the predicted silhouettes and point features only match the observations for very small articulations. In this chapter we present a method for accounting for articulated motion directly within the framework laid out in Chapter 4 and Chapter 5. The equations in this chapter explain how the methodology can be used for any articulated object that is described by a kinematic chain with specific examples given for the da Vinci LND instrument.

6.2 Modelling Articulation with Kinematic Chains

A kinematic chain is the most common method of describing a robot manipulator by dividing it into an assembly of N links or rigid bodies each of which define a coordinate frame $\mathcal{F}$. These links are

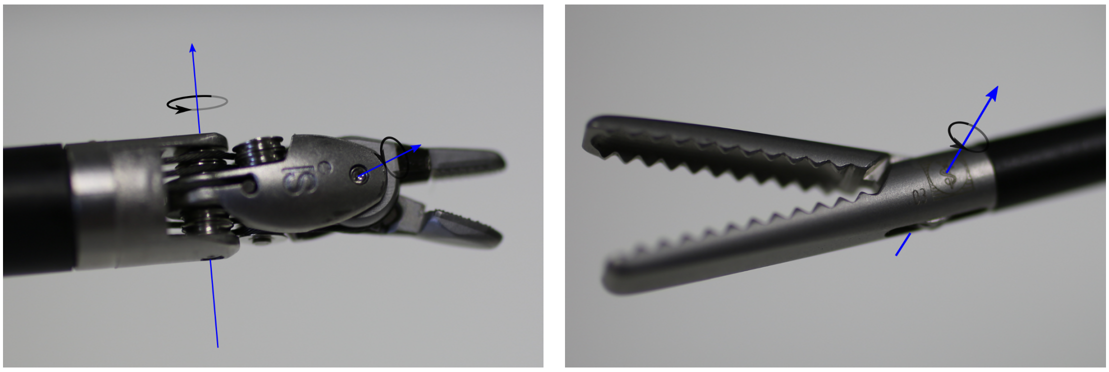

(a) **(b)**

Figure 6.1: (a) A da Vinci LND instrument. This instrument is articulated though rotation of the instrument head, mimicking the motion of a human wrist and additionally the orienting and opening/closing of the claspers. (b) A laparoscopic instrument where the single degree of freedom clasper enables the instrument to open and close.

connected together at a shared axis known as a joint, where for an N link chain there are at most $N-1$ joints. The coordinate frames of consecutive links are related with a single 4×4 transform ${}^{n-1}\mathbf{T}_n$ which is described with one or more DOFs, which specifies how many parameters are required to fully locate the geometry of the connected n^{th} link in the reference frame of the parent $n-1^{\text{th}}$ link [187]. The most common case for robotic manipulators is to use a single DOF joint where the transform is defined to rotate around 1 axis (rotary) or translate along 1 axis (prismatic) and in fact any K DOF joint can be modelled as a series of single DOF joints [188]. In the remainder of this thesis, we shall focus solely on single DOF joints both for simplicity and the fact that the majority of surgical instruments are composed of single DOF joints.

When combined together, the links and joints of a kinematic chain describe how a point $\mathbf{X}$ defined in the local coordinate system of the $j^{\text{th}} : j \leq N$ link $\mathcal{F}_j$ can be transformed into the coordinate system of the base frame of the robot as:

$$\mathbf{X}_0 = {}^{0}\mathbf{T}_1 \; {}^{1}\mathbf{T}_2 ... {}^{j-1}\mathbf{T}_j \mathbf{X}_j \tag{6.1}$$

where ${}^{0}\mathbf{T}_1 \; {}^{1}\mathbf{T}_2 ... {}^{j-1}\mathbf{T}_j$ can be compactly represented as ${}^{0}\mathbf{T}_j$, $\mathbf{X}_j$ is the representation of $\mathbf{X}$ in $\mathcal{F}_j$ and $\mathbf{X}_0$ is the representation of $\mathbf{X}$ in $\mathcal{F}_0$. We drop the subscript $model$ used in previous Chapters denoting the point was defined in the model reference frame for brevity and instead use the index in the kinematic chain. There are several methods to define the transform between neighboring links and for general transforms, 6 DOFs are required to fully specify the relative orientation. However, for single DOF joints, the Denavit Hartenberg (DH) representation [189] defines the n^{th} joint to be parallel to the $x=0$ plane of $\mathcal{F}_{n-1}$, effectively cancelling out 2 degrees of freedom, 1 in rotation and 1 in translation reducing the number of parameters to 4, 2 distances and 2 angles [190]. 1 distance parameter is required to describe how far along the x axis of $\mathcal{F}_{n-1}$ this parallel plane lies and 1 angle parameter describes the rotation between the z axes of $\mathcal{F}_{n-1}$ and $\mathcal{F}_n$ in this plane. These 2 parameters are denoted a_{n-1} and α_{n-1} respectively. Describing how $\mathcal{F}_n$ is orientated relative to its z axis and to $\mathcal{F}_{n-1}$ involves a further 2 parameters. Firstly, the distance along its z axis between where a_{n-1} from link $n-1$ intersects the z axis and where a_n from link n intersects the z axis is defined as d_n and describes the vertical shift between the two links. Additionally, the rotation around the z axis is defined as θ_n. The link length a_n and the the link twist α_n depend on joint axis n and $n+1$ so are defined as 0 for the end of the chain. The link offset d_i and joint angle θ_i are defined between joint 2 and joint $n-1$ and for a revolute joint 1, θ_1 can be chosen arbitrarily and d_1 is set to 0 and if joint 1 is prismatic d_1 is arbitrary and θ_1 is set to

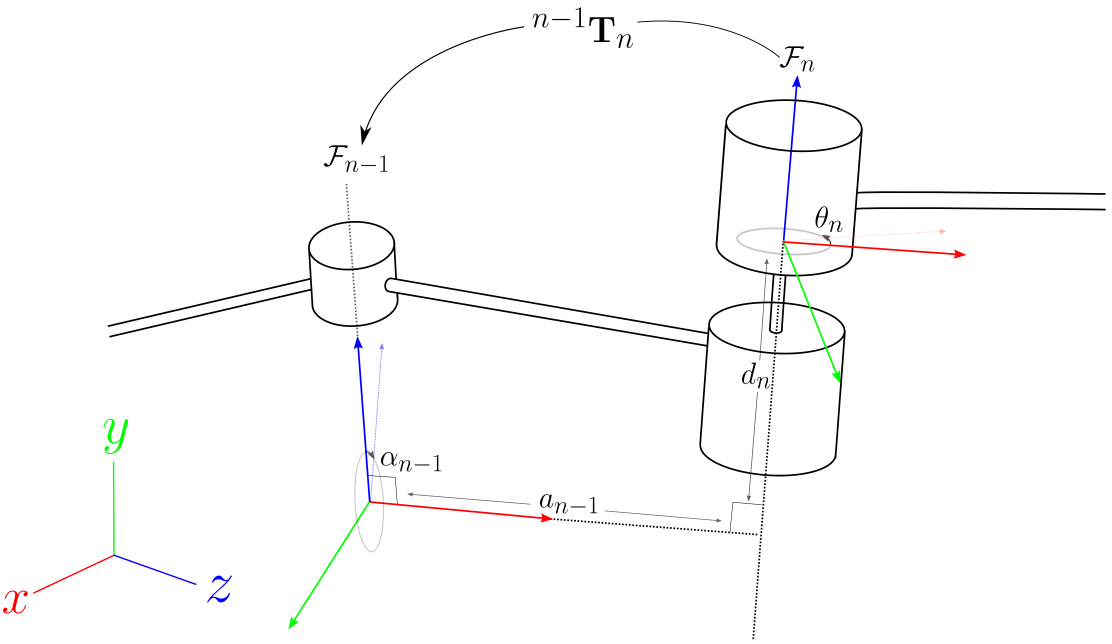


Figure 6.2: The coordinate system transforms used in a modified DH parameter setup. A point defined in the frame $\mathcal{F}_n$ can be transformed into the frame $\mathcal{F}_{n-1}$ with the transform ${}^{n-1}\mathbf{T}_n$.

0 [188]. When applied to a prismatic joint i a_i, α_i, θ_i are fixed and d_i is the DOF whereas for a revolute joint i, a_i, α_i, d_i are fixed and θ_i is the DOF. These 4 rotation and translation operations are applied consecutively to provide a single transform ${}^{n-1}\mathbf{T}_n$ as:

$$
{}^{n-1}\mathbf{T}_n = R_{x_{n-1}}(\alpha_{n-1}) \cdot T_{x_{n-1}}(a_{n-1}) \cdot R_{z_n}(\theta_n) \cdot T_{z_n}(d_n) \tag{6.2}
$$

where $R_{x_{n-1}}$ refers to a 4×4 transform composing a rotation matrix around the x axis of frame $\mathcal{F}_{n-1}$ with a zero translation and R_{z_n} has the same meaning but the rotation component is defined around the z axis of frame $\mathcal{F}_n$. $T_{x_{n-1}}$ and T_{z_n} refer to same concept but the rotation part of the transform is the identity matrix and the translation part is a translation along the x and z axes of frames $\mathcal{F}_{n-1}$ and $\mathcal{F}_n$ respectively. Combined together this forms the single transform:

$$
{}^{n-1}\mathbf{T}_n = \begin{vmatrix} \cos\theta_n & -\sin\theta_n & 0 & a_{n-1} \\ \sin\theta_n \cos\alpha_{n-1} & \cos\theta_n \cos\alpha_{n-1} & -\sin\alpha_{n-1} & -d_n \sin\alpha_{n-1} \\ \sin\theta_{n \sin\alpha_{n-1}} & -\cos\theta_n \sin\alpha_{n-1} & \cos\alpha_{n-1} & -d_n \cos\alpha_{n-1} \\ 0 & 0 & 0 & 1 \end{vmatrix} \tag{6.3}
$$

If each joint is under zero force then the configuration of the robot is exactly defined by the DH parameters of each link. However, when a force is applied to a joint by a motor then, in the case of a rotary joint, a rotation around the z axis of frame n occurs or, in the case of a *prismatic* joint, a translation along the x axis of frame n, resulting in the configuration being defined by $\phi = (a, \alpha, d + \delta d, \theta)$ or $\phi = (a, \alpha, d, \theta + \delta\theta)$. The DH parameters for each joint of a given instrument are normally obtained from the manufacturer. DH parameters are typically separated into classic and modified representations and in this thesis, we focus on the modified parameters which attaches frame $\mathcal{F}_i$ to link i and places the origin of $\mathcal{F}_i$ on joint axis i.

6.3 DH Parameters for Da Vinci Robotic Instruments

In this chapter we focus solely on working with the instruments of the da Vinci robotic system, particularly the LND instrument which is commonly used in surgical procedures to control a suturing needle. However, the methods are easily applicable to any robotic instrument with the appropriate minor modifications. The LND, like any da Vinci instrument, has 3 DOFs on the wrist: firstly, the wrist pitch (WP) which articulates the entire wrist to mimic the motion of a human wrist enabling the mirroring of motions such as stitching to be captured more precisely. The second DOF is the wrist yaw (WY) which corresponds to a coordinated motion of two mechanical joints representing the claspers and enables the claspers to be oriented towards a target. The final DOF allows the clasper to open and close so that the instrument can grasp and hold objects.

	Joint Type	**Arm Index**	$\mathbf{a_{i-1}}$ (m)	$\boldsymbol{\alpha_{i-1}}$ (rads)	$\mathbf{d_i}$ (m)	$\boldsymbol{\theta_i}$ (rads)
Wrist Pitch	Rotary Joint	5	0	$-\frac{\pi}{2}$	0	$-\frac{\pi}{2}$
Wrist Yaw	Rotary Joint	6	0.009	$-\frac{\pi}{2}$	0	$-\frac{\pi}{2}$
Grip	Rotary Joint	7	0	$-\frac{\pi}{2}$	0	0

Table 6.1: Large Needle Driver DH parameters for the articulated wrist. The arm index refers to the actual joint location in the full 7 DOF da Vinci arm. The first 4 DOFs are shown in Figure 6.3.

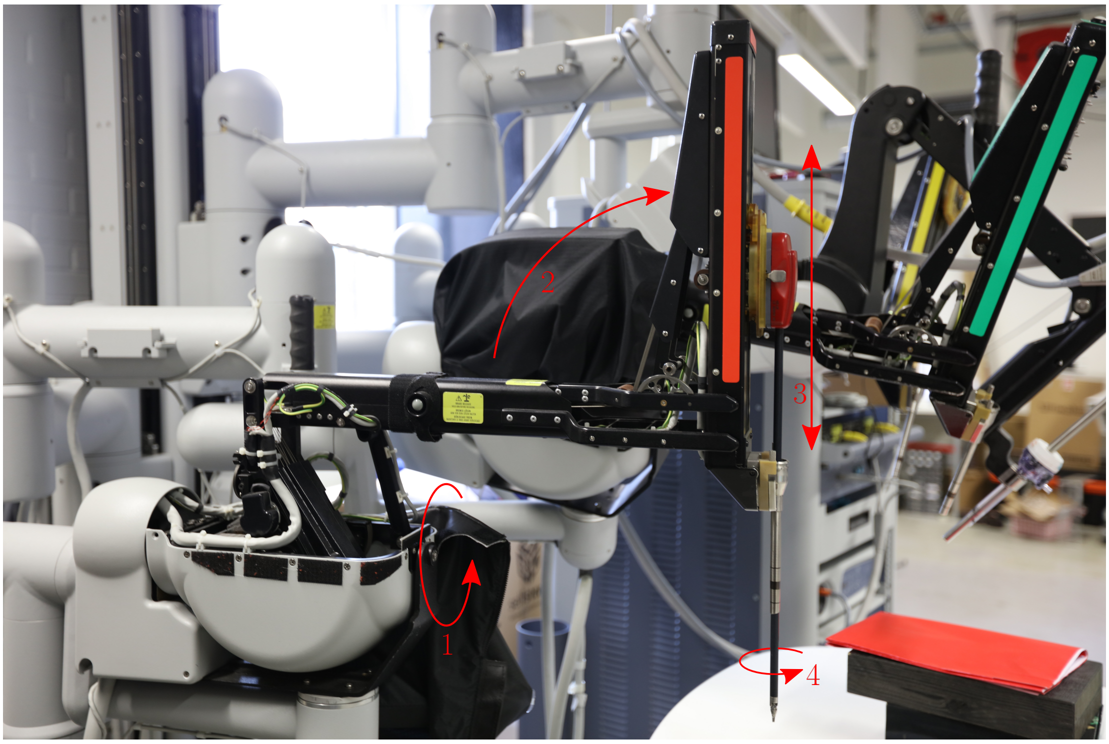
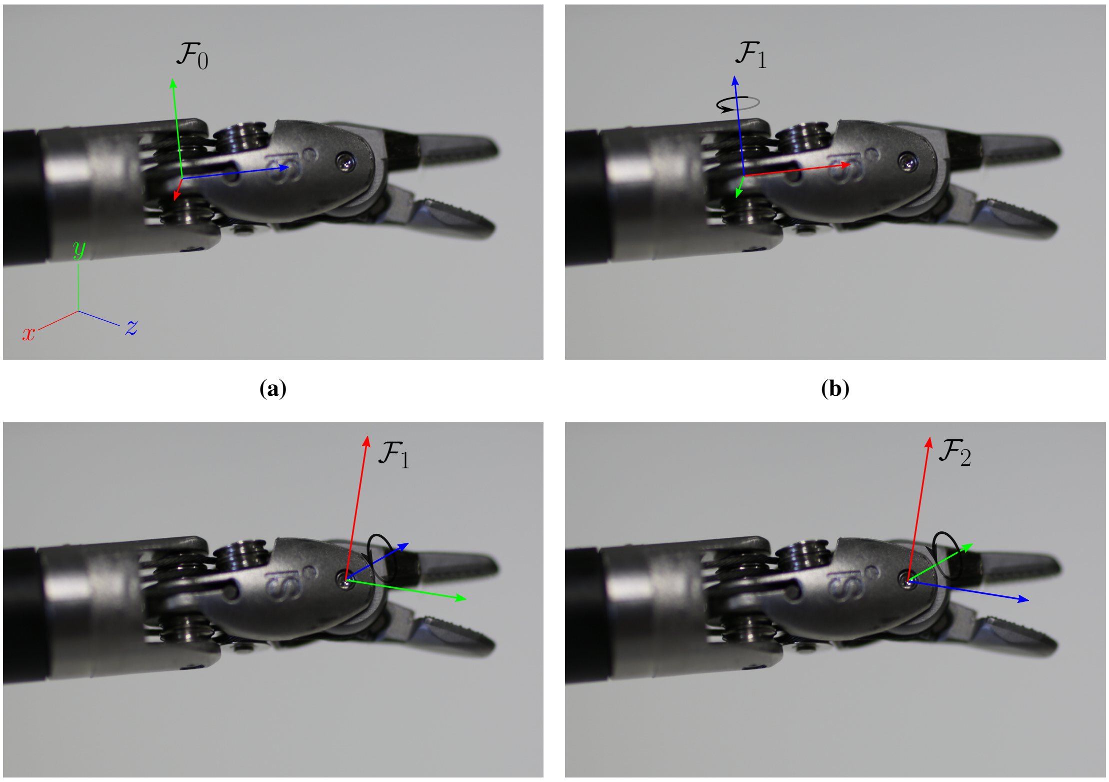


Figure 6.3: The PSM of a da Vinci robot with a LND instrument attached. The first 4 joints of the PSM are labeled, where 1 and 2 provide rotational positioning of the PSM around the remote center of motion (RCM), 3 provides translation along the axis of the instrument, in and out of the patient and 4 allows the instrument to roll on its axis.

Although the kinematics of the da Vinci robot arm involve 7 degrees of freedom for the PSM and a further 6 for the setup joints (SUJ), as we only observe the instrument shaft with the surgical camera, we effectively ignore the DOFs of the SUJ and the first 4 DOFs of the PSM and instead model the orientation of the instrument with a 6 DOF rigid Euclidean transform (see chapter 4). Our model is therefore parameterized with a vector $\boldsymbol{\theta}$ which contains the 6 rigid body parameters of 3D transforms



(a) **(b)**

(c) **(d)**

Figure 6.4: (a) The base frame $\mathcal{F}_0$ for the robotic instrument which is oriented relative to the surgical camera with the rigid body transform ${}^{cam}\mathbf{T}_{model}$. (b) The wrist frame $\mathcal{F}_1$ which enables the instrument head to rotate around the z axis of this frame. (c) The claspers rotate together around the z axis of $\mathcal{F}_1$ defining a new frame $\mathcal{F}_2$ which has its x axis pointing in the direction of the claspers. (d) The claspers rotate around the z axis of this frame in opposite directions allowing opening and closing.

and 3 articulated DOFs which are defined by the kinematic structure of the robotic instrument:

$$\boldsymbol{\theta} = (t_x, t_y, t_z, r_x, r_y, r_z, a_1, a_2, a_3) \tag{6.4}$$

where $\mathbf{t} = (t_x, t_y, t_z)$ is the translation vector between the origin of the camera coordinate system and the instrument coordinate system, (r_x, r_y, r_z) are the rotation Euler angles around the (x, y, z) axis of the coordinate system and (a_1, a_2, a_3) are the parameters with define motion around or along each of the joints of the kinematic chain of the instrument.

6.4 Integrating Articulation into the Tracking Framework

A particular advantage of working with a region based framework is that articulation can be easily integrated through rendering of the silhouette at each new pose configuration and adding the additional degrees of freedom into the optimization over the cost defined by the SDF ϕ of the silhouette. When we compute the the region based cost function in Equation 4.14, each pixel $\mathbf{x}$ is assigned to a particular vertex $\mathbf{X}$ on the model surface via the transform $\mathbf{x} = \mathbf{K}\mathbf{T}(\boldsymbol{\theta})\mathbf{X}$ and to compute the Jacobian we must know $\mathbf{X}$ and $\mathbf{T}$. This is trivial when we work with a fully rigid model, as $\mathbf{T} = {}^{cam}\mathbf{T}_{model}$ for each vertex and $\mathbf{X}$ is computed with simple backprojection. However when the model is articulated we need to know which of the model frames $\mathcal{F}_i$ the vertex belongs to so that we can compute ${}^{cam}\mathbf{T}_i$ via forward kinematics and $\mathbf{X}_i$ via backprojection. In our method, we achieve this by assigning a model component index to each pixel in the image when we render the silhouette which corresponds to the numerical index of the corresponding vertex's frame. For background pixels which do not directly project to a model vertex, we perform a brute force lookup for the closest frame. With the closest reference frame $\mathcal{F}_i$ and transform ${}^{0}\mathbf{T}_i$ we can compute the derivative of the vertex with respect to each parameter. The Jacobian of the frame to camera transform part of this equation breaks down as:

$$\frac{\partial^{cam}\mathbf{T}_i(\boldsymbol{\theta})\mathbf{X}_i}{\partial\theta_j} = \frac{\partial}{\partial\theta_j}\, {}^{cam}\mathbf{T}_0\ {}^{0}\mathbf{T}_{j-1}\ {}^{j-1}\mathbf{T}_j\ {}^{j}\mathbf{T}_i\mathbf{X}_i \tag{6.5}$$

where ${}^{j-1}\mathbf{T}_j$ is the transform from the parent of frame $\mathcal{F}_j$ to $\mathcal{F}_j$. If we consider the parameter θ_j as being responsible for rotating the jth link around the z axis of its frame (see Section 6.2), then the derivative becomes:

$$\begin{aligned}\frac{\partial^{cam}\mathbf{T}_i(\boldsymbol{\theta})\mathbf{X}_i}{\partial\theta_j} &= {}^{cam}\mathbf{T}_0\ {}^{0}\mathbf{T}_{j-1}\left(\frac{\partial}{\partial\theta_j}\,{}^{j-1}\mathbf{T}_j\right){}^{j}\mathbf{T}_i\mathbf{X}_i && (6.6)\\ &= {}^{cam}\mathbf{T}_0\ {}^{0}\mathbf{T}_{j-1}\left(\mathbf{z}\times\mathbf{X}_j\right) && (6.7)\end{aligned}$$

where the product rule is applied to each transform of the kinematic chain and, as each parameter influences directly only a single $\mathbf{T}$, all but a single term is zero. The vertex $\mathbf{X}_i$ is effectively transformed into the coordinate frame $\mathcal{F}_j$ as this equation measures how motion of the frame j influences vertices in frames towards the distal end of the kinematic chain. When optimizing the parameters of a rigid object each point on the contour contributes to the pose update for the entire model. However, when working with an articulated instrument, there are parts of the contour which only affect a single articulated component. For instance, for a da Vinci instrument misalignments in the wrist part of the instrument will not be affected by modifying the articulated parameters of the claspers and hence this part will have zero contribution to the Jacobian.

Integrating articulated DOF tracking into the point tracking framework similarly involves back projecting each tracked point onto the model surface and storing its 3D location in the frame $\mathcal{F}_i$ of the i^{th} joint on the articulated instrument. The cost is applied as in Equation 5.1 and the Jacobian of the point with respect to the pose parameters is trivially extended with Equation 6.6. As this framework is often applied to robotic surgical instruments where full axis roll is a common occurrence we also need a method of regenerating optical flow features if the instrument rotates around its central axis to an angle where few of the original points are visible. We deem this situation has occurred when less than 4 of the tracked points are predicted to be visible and if this occurs, we regenerate the points using the current estimate of pose. A particular disadvantage of this technique is that drift in the model pose can accumulate over time if the frame at which the points are regenerated has an inaccurate estimate of pose. We also provide a further modification to better handle the common axial rotation of the instrument. Although our optical flow tracked points roll the instrument around its central axis through the Jacobian of the point projection, this term is often small relative to the 2D translation term provided by this derivative and as we compute the rotation Jacobian as a quaternion, there is no way to artificially scale this parameter. To account for this, we perform a 3 step brute force roll rotation check after convergence for each frame where we incrementally rotate the instrument by $0.03, 0.06$ and 0.09 radians in the positive and negative direction and update the instrument roll rotation to the value which gives the smallest summed reprojection error across all of the tracked points.

6.5 Online Forest Learning

To improve the quality of the segmentation used to drive the level set based pose estimation, which is much more sensitive to noise than the rigid pose estimation, we can make two improvements to the random forest training. Firstly, as we only wish to classify the background and foreground in regions near the model contour, it makes sense to learn a highly specific model for the appearance using only pixels which sit close to this boundary. As we have a full 3D model of the instrument, we can generate automatic ground truth segmentations from the signed distance function and only using pixels within a predefined threshold in the training set. After a specific number of frames, we retrain the forest although this in principal could be performed in a background thread to minimize performance overhead. Our preliminary experiments showed that the most effective masks were obtained from learning both the foreground and background model from the first frame and updating the background model in subsequent frames from the data in the first frame, where the current estimate of pose is used to select background pixels. This prevents model drift from affecting the training data significantly by incorrectly placing background pixels into the foreground class and vice-versa, however a limitation of this method is that if the camera is moved significantly this can impact the color distribution of expected background pixels. To alleviate this, a system which detects camera movement and reinitializes the background image would be required, which for instance could be achieved using the robot control system in RMIS.

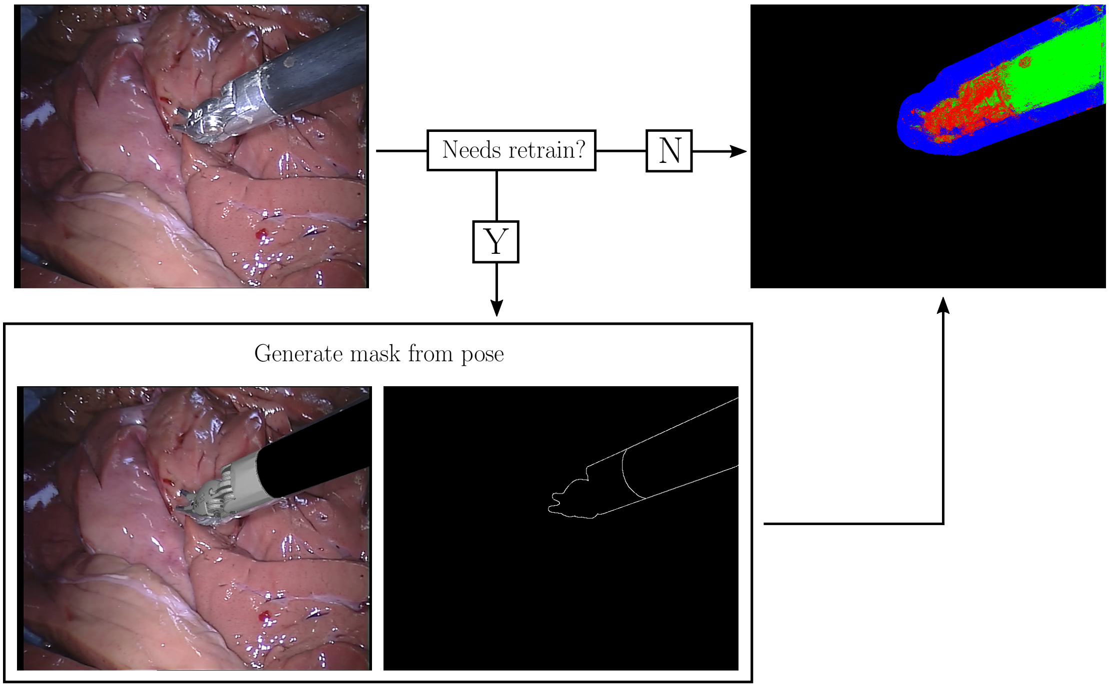


Figure 6.5: The online forest algorithm. For each new frame, we check if the forest needs to be re-learned and generate a new ground truth mask from the projection of the estimate of pose onto the first frame. We only use pixels from a fixed size region around the contour to generate background samples and use all of the pixels within the contour to generate foreground samples. Once a new model is learned, this is then applied to each subsequent frame until re-training is again required.

6.6 Experiments

To evaluate the accuracy of the articulated tracking we again perform quantitative ex-vivo studies. However, as several recently published methods of articulated instrument tracking provide comparison ex-vivo and in-vivo datasets, we can also perform a quantitative comparison with these methods. This comparison was not appropriate for chapter 4 as these methods are validated on highly articulated sequences which the rigid instrument would be unable to model effectively.

6.6.1 Implementation Details

Our implementation is based on the OpenGL/GLSL implementation of rigid tool tracking discussed in Chapter 4 however, we instead describe our model as a tree of nodes in a parent-child relationship. For the example da Vinci LND model, this consists of a base frame containing the shaft which has a single child node containing the wrist model. This again has a single child node containing the clasper axis but no geometry which in turn has two child nodes containing each clasper. Each node maintains the DH parameters describing the transform to its parent and this is used to render the components relative to one another.

During our experiments we noticed a particular error occurring when a clasper could effectively become lost due to noise in the background misaligning it from the visual data (see Figure 6.21). As the region based pose alignment requires at least some overlap for the model to converge, this becomes a particular problem. We built a recovery system into the claspers which effectively attempts to search for the clasper by gradually closing and reorienting the pair, keeping the correctly positioned clasper static, until either the misaligned clasper becomes realigned or they close. As the surgeon often closes the claspers during a procedure, this process effectively recovers the situation for us.

6.6.2 Ex-vivo Experiments

We construct identical ex-vivo experiments to Chapter 4 using the da Vinci LND instrument and several different animal tissue samples. The camera maintains a static position and observes 1000 frame sequences showing an instrument moving with articulation of the wrist and claspers. The DVRK platform is used to capture synchronized joint and video data and we use the same manual joint correction techniques to obtain a more accurate ground truth. To provide a comparison estimate we use the uncorrected output from the joint encoders and forward kinematics.

The results are quite consistent across all of our datasets, with the best performance obtained in dataset 2 where trajectory plots are shown in Figures 6.10 and 6.11 and qualitative results are shown in Figures 6.12 and 6.13. This dataset is visually similar to dataset 4 in Chapters 4 and 5 and in both cases is our best performing dataset. This is due to the extremely clear discrimination between the instrument and the liver tissue surface leading to extremely accurate silhouettes. Datasets 1 and 4 both contain a line of fatty tissue across the center of the frame (see Figures 6.16, 6.17, 6.20 and 6.21) which confuses the clasper pose estimation due to visually similar color between the two. The clasper is the most vulnerable part of the instrument to this type of error as it is much smaller than the head and the shaft and so cannot rely on other well classified parts of the instrument model.

Dataset	$t_x(mm)$	$t_y(mm)$	$t_z(mm)$	$r_x(rads)$	$r_y(rads)$	$r_z(rads)$	wrist (rads)	grasper (rads)	opening (rads)
Dataset1 - MR LK.	2.62 ± 1.95	1.55 ± 1.28	15.51 ± 7.04	0.48 ± 0.33	0.12 ± 0.11	0.09 ± 0.07	0.23 ± 0.13	0.23 ± 0.20	0.27 ± 0.33
Dataset1 - R	10.07 ± 1.05	7.49 ± 0.80	2.93 ± 1.37	0.04 ± 0.02	0.01 ± 0.01	0.03 ± 0.01	0.03 ± 0.02	0.07 ± 0.03	0.08 ± 0.07
Dataset2 - MR LK.	1.32 ± 1.35	0.89 ± 1.05	3.55 ± 1.67	0.28 ± 0.14	0.06 ± 0.03	0.06 ± 0.04	0.10 ± 0.06	0.13 ± 0.12	0.15 ± 0.16
Dataset2 - R	10.11 ± 1.00	8.68 ± 0.74	1.40 ± 0.54	0.14 ± 0.02	0.01 ± 0.01	0.04 ± 0.01	0.03 ± 0.03	0.04 ± 0.03	0.08 ± 0.06
Dataset3 - MR LK.	1.24 ± 1.09	0.86 ± 0.69	7.91 ± 3.99	0.31 ± 0.18	0.12 ± 0.09	0.09 ± 0.07	0.28 ± 0.24	0.32 ± 0.21	0.37 ± 0.35
Dataset3 - R	9.67 ± 0.91	7.50 ± 1.01	4.00 ± 2.07	0.04 ± 0.00	0.01 ± 0.00	0.02 ± 0.00	0.00 ± 0.00	0.11 ± 0.08	0.03 ± 0.03
Dataset4 - MR LK.	1.76 ± 0.98	0.78 ± 0.62	5.87 ± 3.29	0.12 ± 0.11	0.11 ± 0.09	0.06 ± 0.05	0.30 ± 0.20	0.37 ± 0.23	0.19 ± 0.19
Dataset4 - R	10.36 ± 1.33	7.75 ± 1.34	3.69 ± 2.87	0.14 ± 0.08	0.14 ± 0.02	0.04 ± 0.02	0.11 ± 0.08	0.15 ± 0.13	0.21 ± 0.21
Mean error MR LK.	1.74 ± 1.34	1.02 ± 0.91	8.21 ± 4.00	0.30 ± 0.19	0.10 ± 0.08	0.08 ± 0.06	0.23 ± 0.16	0.26 ± 0.19	0.25 ± 0.26
Mean error R	10.05 ± 1.07	7.86 ± 0.97	3.00 ± 1.71	0.09 ± 0.03	0.04 ± 0.01	0.03 ± 0.01	0.04 ± 0.03	0.09 ± 0.06	0.10 ± 0.09

Table 6.2: Errors for 3D articulated pose estimation for our tracking method (MR LK) compared with the uncorrected kinematics (R) against the hand corrected pose estimates. The mean translation, rotation and articulation errors ± the standard deviation over all frames are shown for each dataset. The last two rows show the overall error over all datasets. The overall results show that the robotic system is very inaccurate in the t_x and t_y degrees of freedom but much more accurate over other degrees of freedom. The visual method struggles heavily with t_z in comparison and is slightly more inaccurate with rotational degrees of freedom. The articulated parameters are estimated almost perfectly by the robotic system, as the kinematic inaccuracies caused the cable driven joints are less influential in these degrees of freedom as there are fewer joints influencing these measurements.

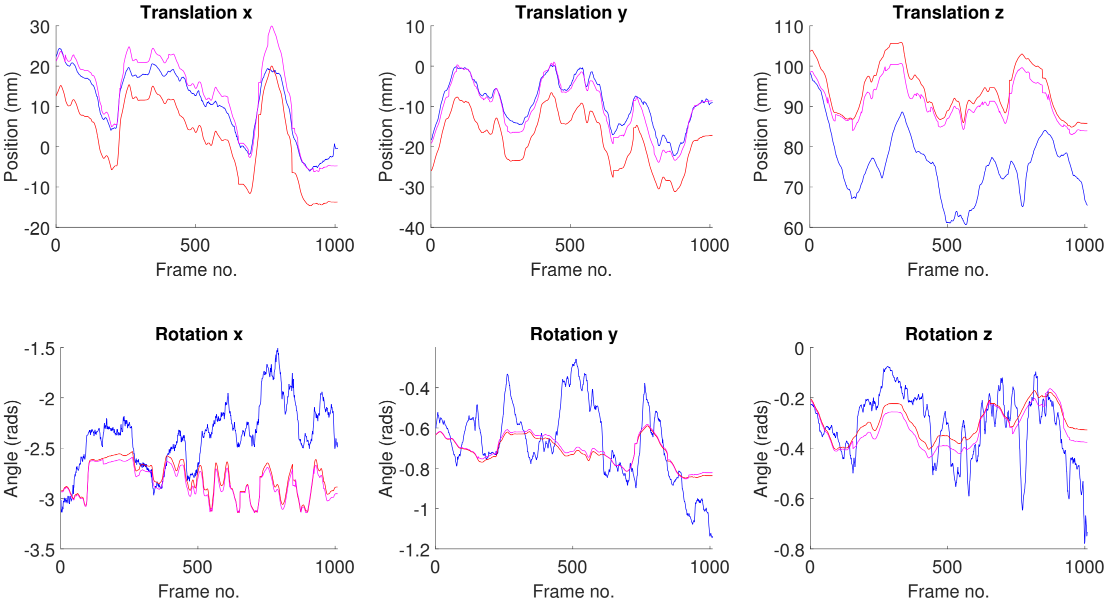


Figure 6.6: Robot Kinematics, MR LK Tracker, Ground Truth. Quantitative analysis of the articulated tracking results for dataset 1 compared with the robot kinematics. The top row shows the translation trajectories for our tracker and the kinematics compared with the hand corrected ground truth and similarly row 2 shows the rotation trajectories for our tracker and the kinematics compared with the hand corrected ground truth. There are some large rotation errors using the MR LK tracker and around 15 mm of t_z error. The t_y error increases and decreases over the sequence which occurs as the instrument converges to the correct pose and then loses tracking.

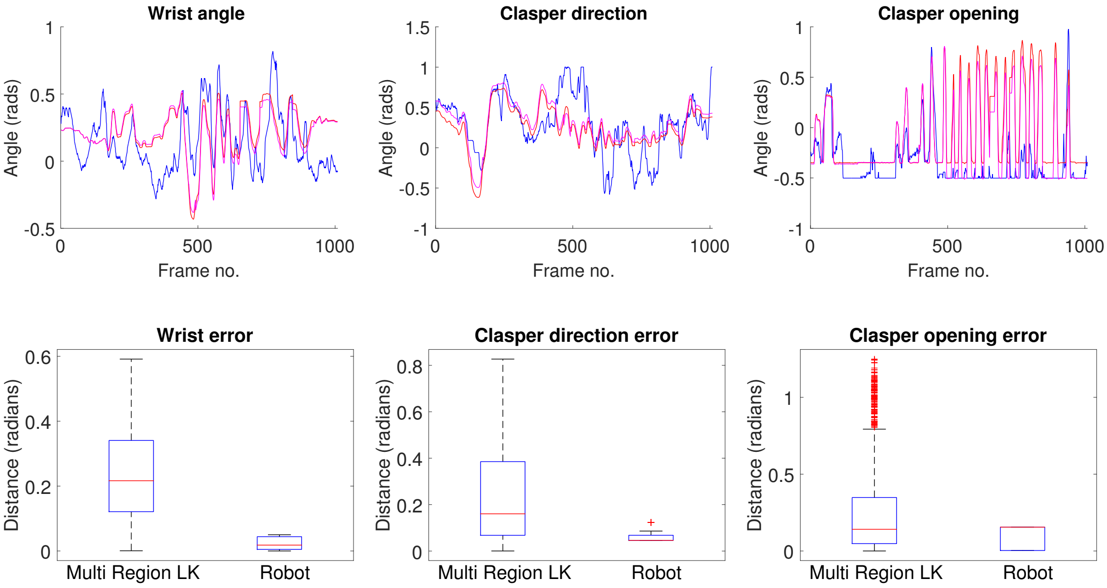


Figure 6.7: Robot Kinematics, MR LK Tracker, Ground Truth. Quantitative analysis of the articulated tracking results for dataset 1 compared with the robot kinematics. Row 1 shows the trajectories for the ground truth, robot kinematics and our tracker for each of the 3 articulated DOFs possessed by the da Vinci instruments. Row 2 shows error distributions for the same DOFs where the red line shows the median error while the top and bottom of the box show the 25th and 75th percentiles, the whiskers extend to the most extreme data points not considered outliers and outliers are plotted individually. The wrist error is particularly high, often missing larger motions.

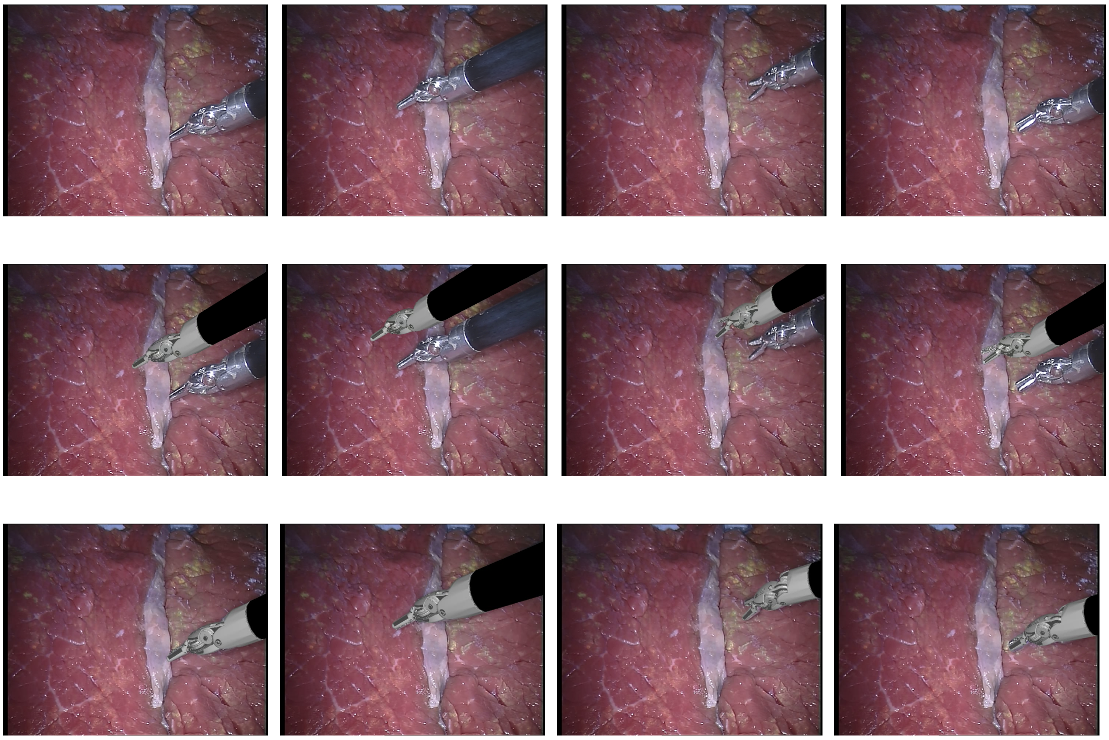


Figure 6.8: Qualitative analysis from ex-vivo dataset 1 showing frames 100, 200, 350 and 400. The top row shows the original frames, the middle row shows the output from the raw, uncorrected kinematics and the bottom row shows the MR LK tracker.

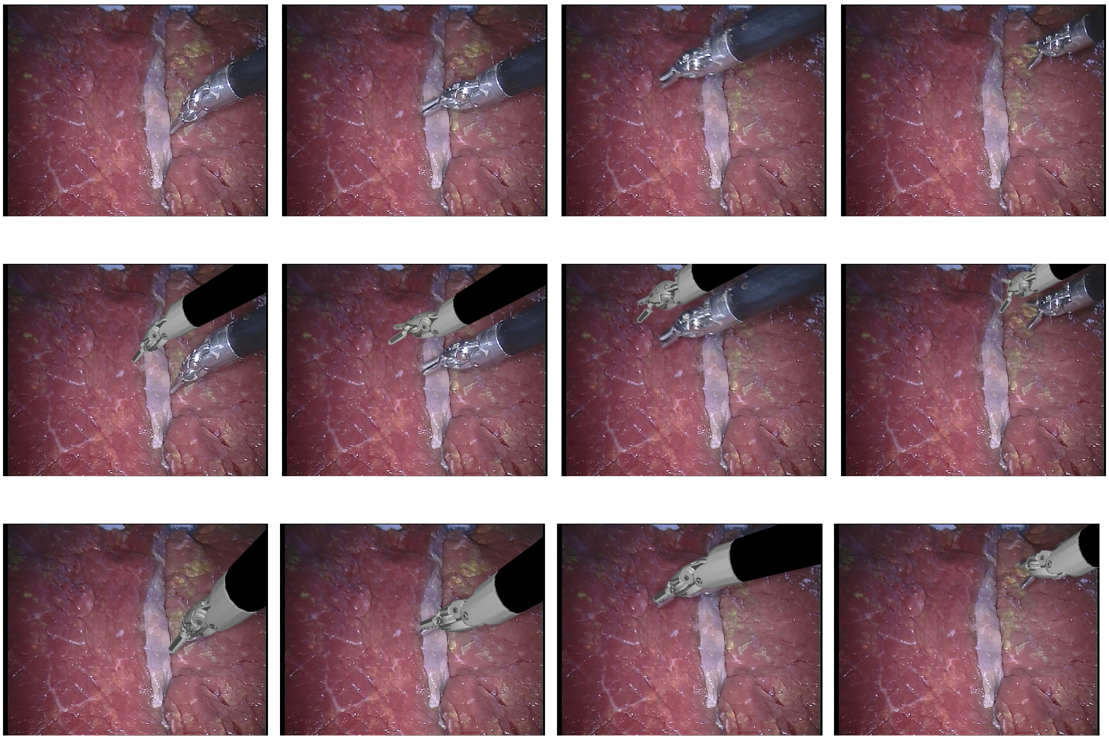


Figure 6.9: Qualitative analysis from ex-vivo dataset 1 showing frames 500, 600, 700 and 1000. The top row shows the original frames, the middle row shows the output from the raw, uncorrected kinematics and the bottom row shows the MR LK tracker.

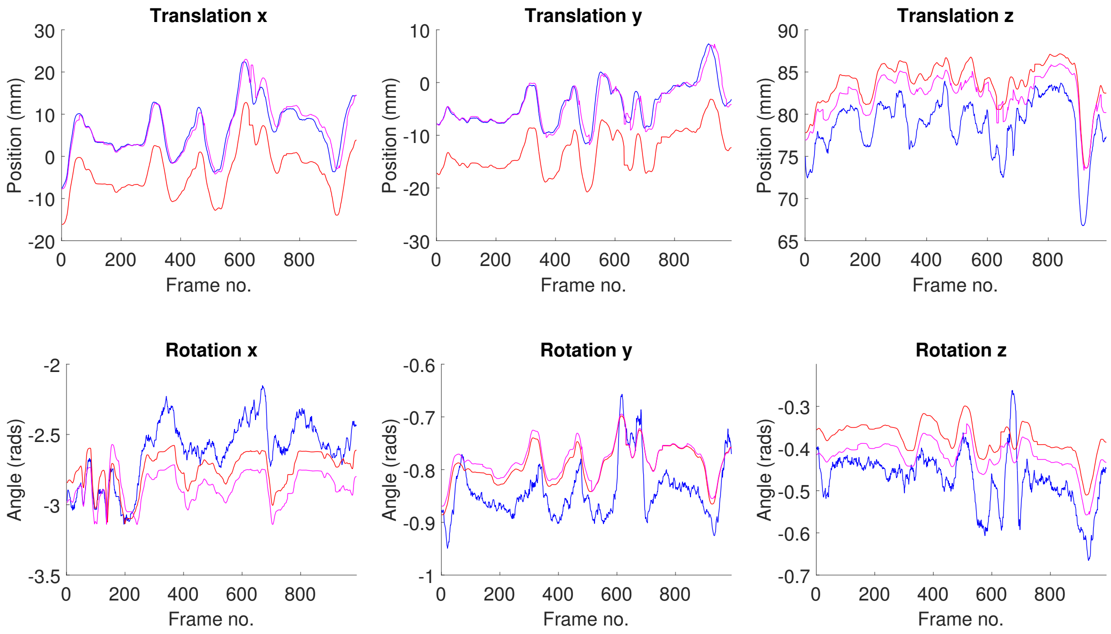


Figure 6.10: Robot Kinematics, MR LK Tracker, Ground Truth. Quantitative analysis of the articulated tracking results for dataset 2 compared with the robot kinematics. The top row shows the translation trajectories for our tracker and the kinematics compared with the hand corrected ground truth and similarly row 2 shows the rotation trajectories for our tracker and the kinematics compared with the hand corrected ground truth. The MR LK tracker is very accurate over this sequence, due to the excellent color classification against the clean background.

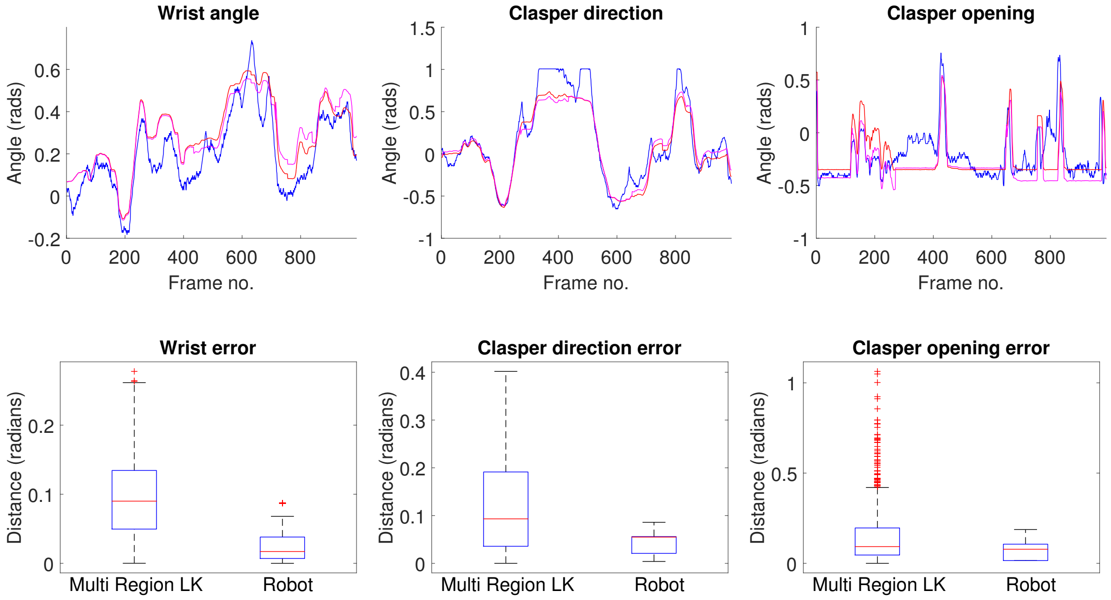


Figure 6.11: Robot Kinematics, MR LK Tracker, Ground Truth. Quantitative analysis of the articulated tracking results for dataset 2 compared with the robot kinematics. Row 1 shows the trajectories for the ground truth, robot kinematics and our tracker for each of the 3 articulated DOFs possessed by the da Vinci instruments. Row 2 shows error distributions for the same DOFs where the boxes have the same meaning as in Figure 6.15.

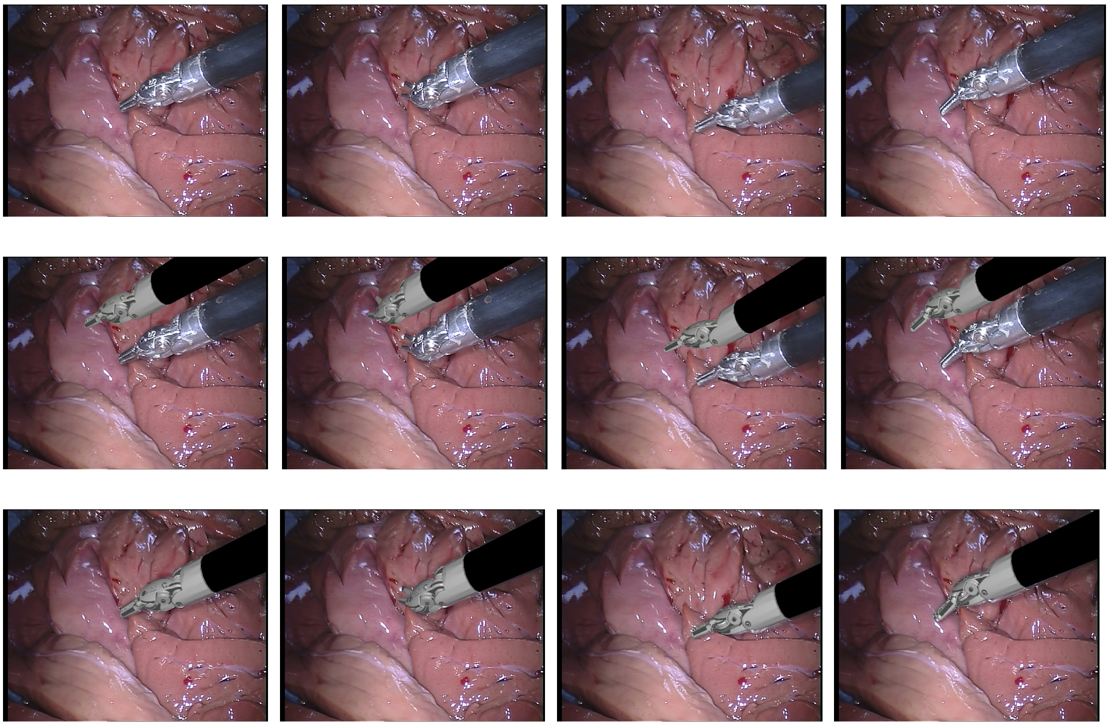


Figure 6.12: Qualitative analysis from ex-vivo dataset 2 showing frames 100, 200, 300 and 400. The top row shows the original frames, the middle row shows the output from the raw, uncorrected kinematics and the bottom row shows the MR LK tracker. In frame 200, the instrument head rotates in and out of view (see Figure 6.11 and the MR LK method correctly tracks this.

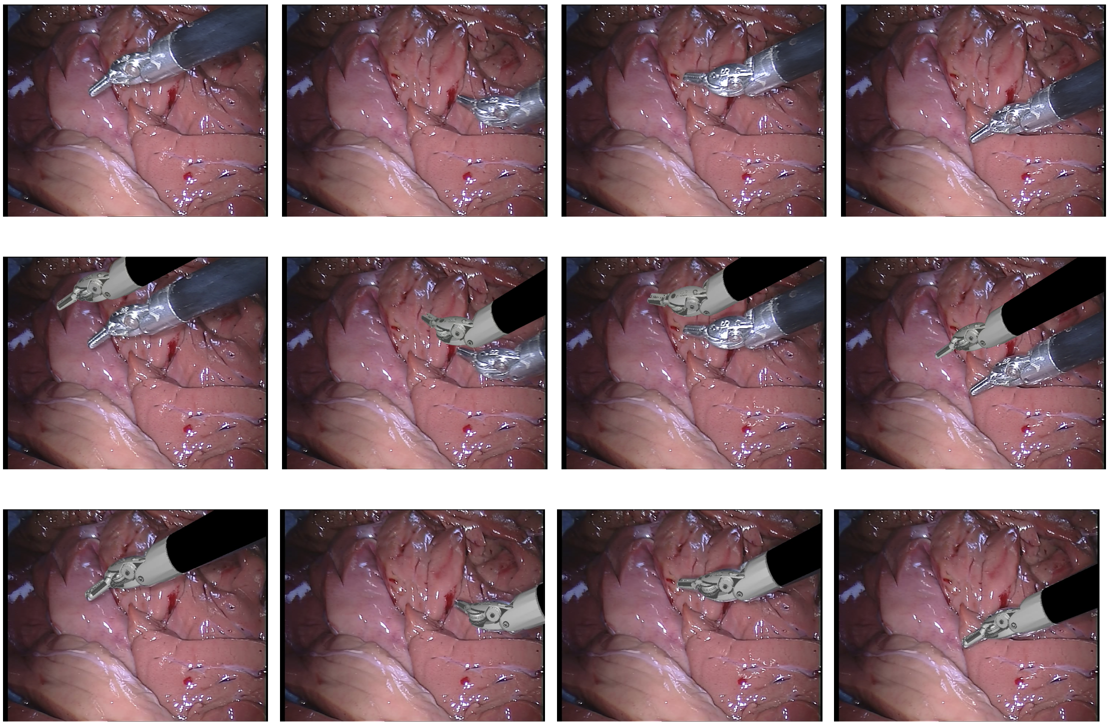


Figure 6.13: Qualitative analysis from ex-vivo dataset 2 showing frames 500, 600, 700 and 1000. The top row shows the original frames, the middle row shows the output from the raw, uncorrected kinematics and the bottom row shows the MR LK tracker. The MR LK tracker retains high accuracy over this sequence.

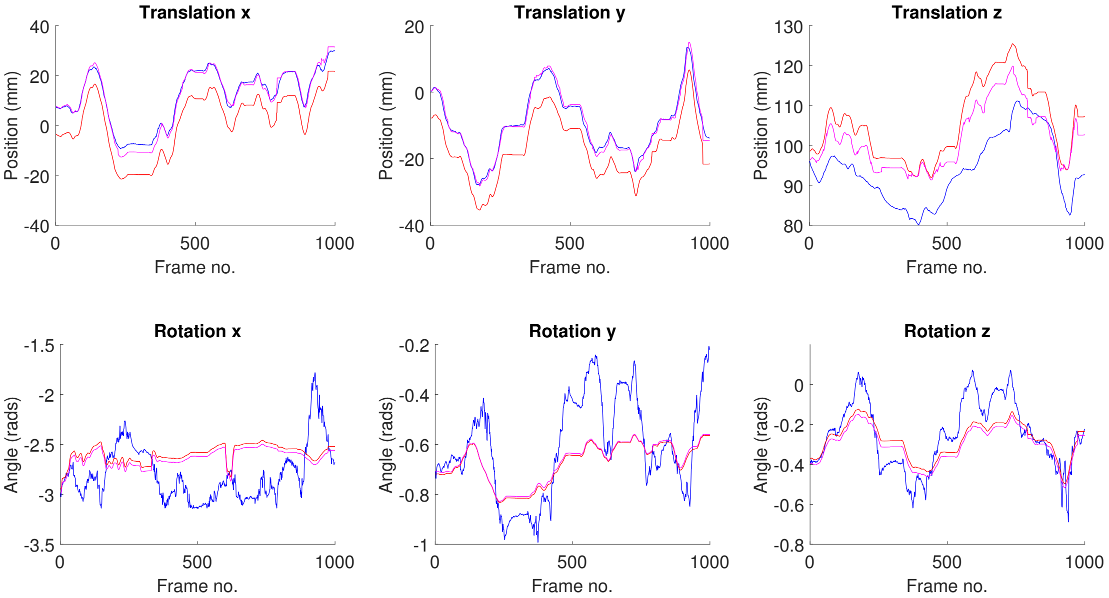


Figure 6.14: Robot Kinematics, MR LK Tracker, Ground Truth. Quantitative analysis of the articulated tracking results for dataset 3 compared with the robot kinematics. The top row shows the translation trajectories for our tracker and the kinematics compared with the hand corrected ground truth and similarly row 2 shows the rotation trajectories for our tracker and the kinematics compared with the hand corrected ground truth. The error for translation is quite low, but larger errors are seen in r_x and r_y which can be explained by the larger t_z values of the sequence which make tracking more challenging.

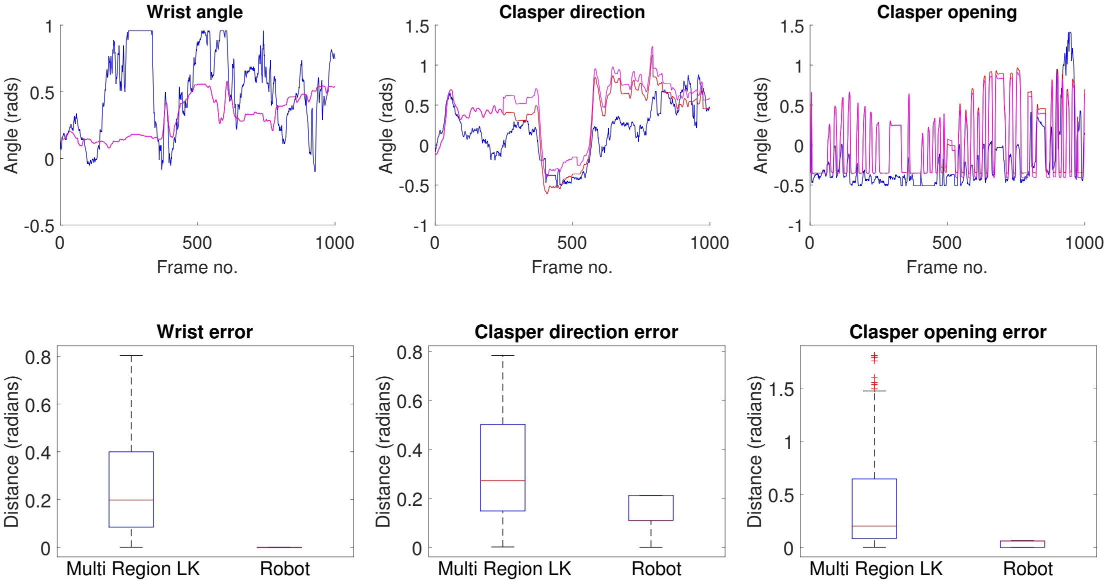


Figure 6.15: Robot Kinematics, MR LK Tracker, Ground Truth. Quantitative analysis of the articulated tracking results for dataset 3 compared with the robot kinematics. Row 1 shows the trajectories for the ground truth, robot kinematics and our tracker for each of the 3 articulated DOFs possessed by the da Vinci instruments. Row 2 shows error distributions for the same DOFs where the boxes have the same meaning as in Figure 6.15.

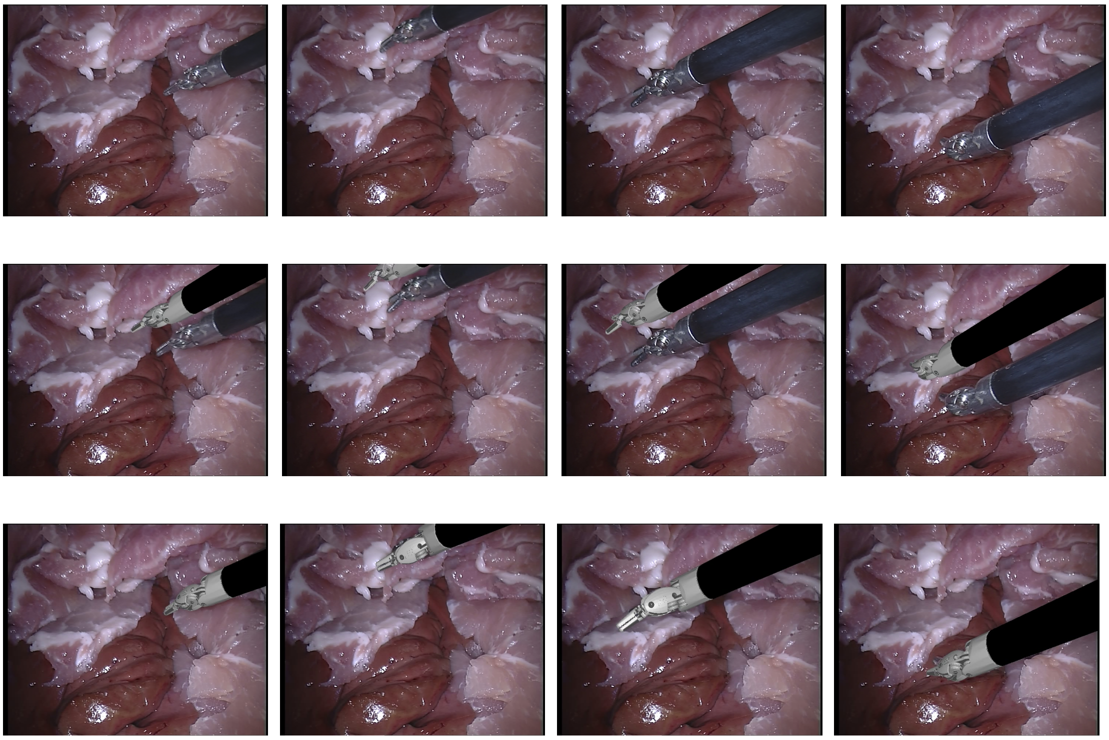


Figure 6.16: Qualitative analysis from ex-vivo dataset 3 showing frames 100, 200, 300 and 400. The top row shows the original frames, the middle row shows the output from the raw, uncorrected kinematics and the bottom row shows the MR LK tracker. Frames 200 and 300 show large inaccuracies in the instrument head rotation.

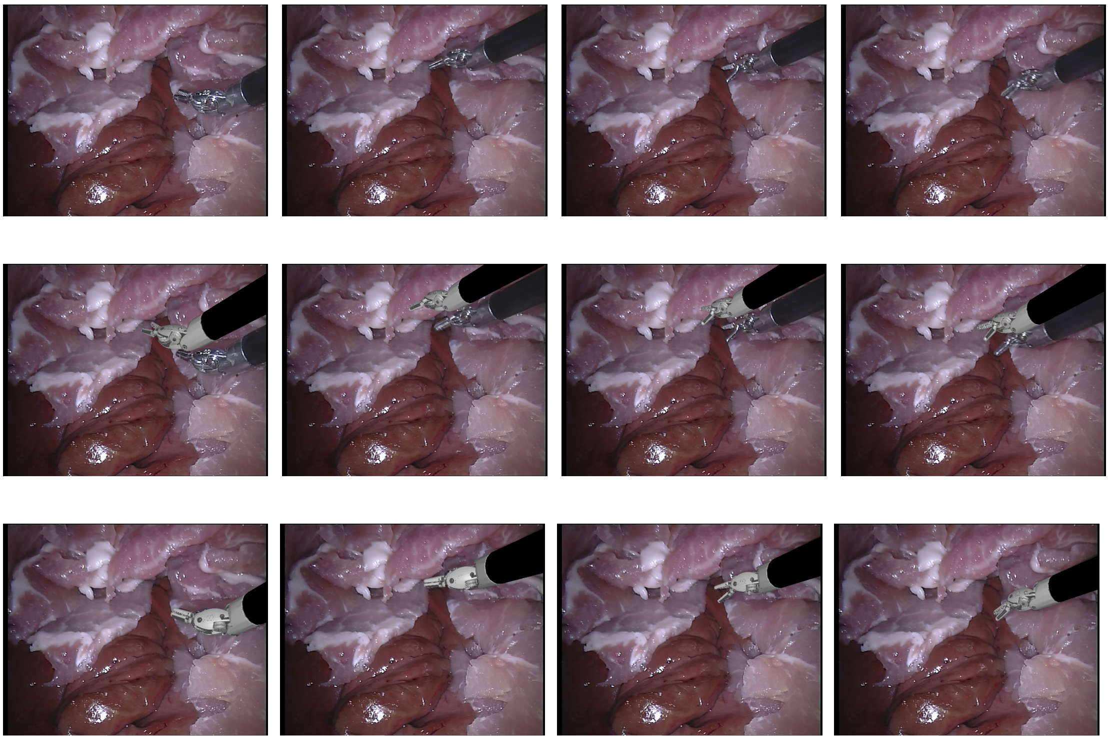


Figure 6.17: Qualitative analysis from ex-vivo dataset 3 showing frames 500, 600, 700 and 1000. The top row shows the original frames, the middle row shows the output from the raw, uncorrected kinematics and the bottom row shows the MR LK tracker. Frame 700 shows a large error in the MR LK estimate of the head angle.

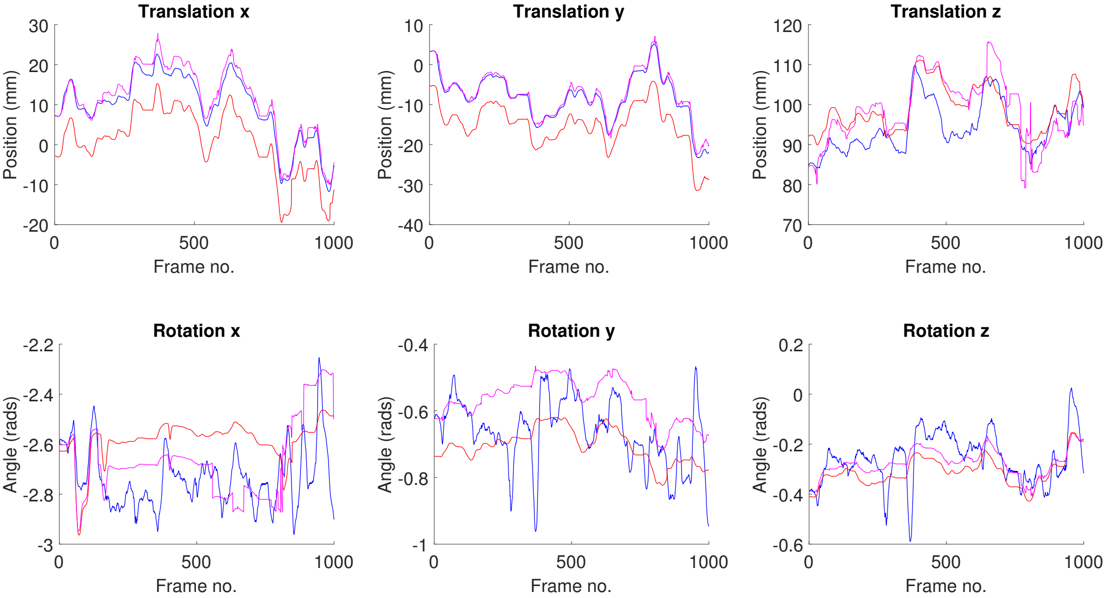


Figure 6.18: Robot Kinematics, MR LK Tracker, Ground Truth. Quantitative analysis of the articulated tracking results for dataset 4 compared with the robot kinematics. The top row shows the translation trajectories for our tracker and the kinematics compared with the hand corrected ground truth and similarly row 2 shows the rotation trajectories for our tracker and the kinematics compared with the hand corrected ground truth.

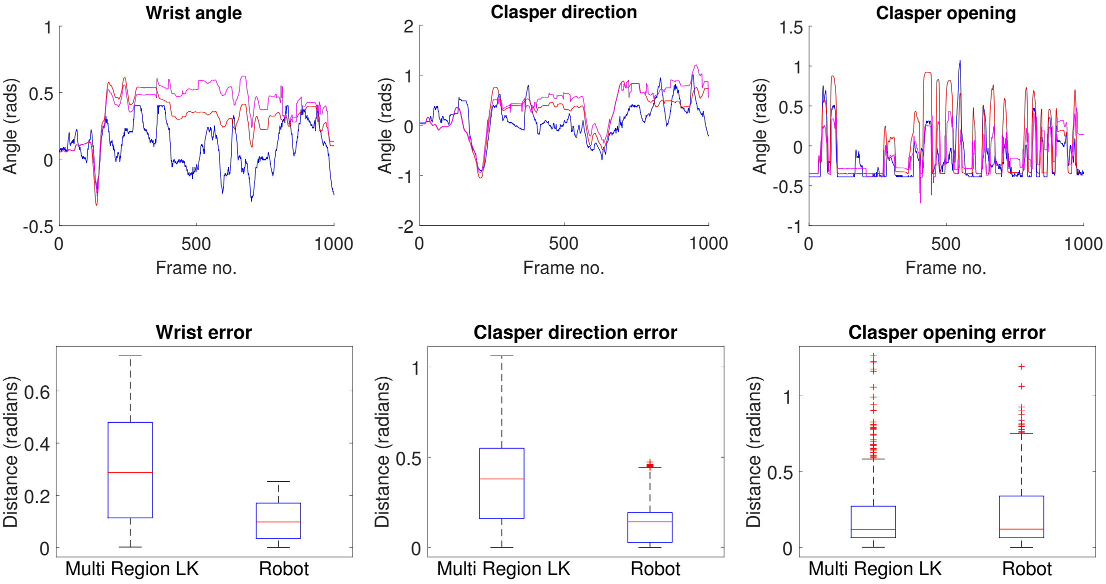


Figure 6.19: Robot Kinematics, MR LK Tracker, Ground Truth. Quantitative analysis of the articulated tracking results for dataset 4 compared with the robot kinematics. Row 1 shows the trajectories for the ground truth, robot kinematics and our tracker for each of the 3 articulated DOFs possessed by the da Vinci instruments. Row 2 shows error distributions for the same DOFs where the boxes have the same meaning as in Figure 6.15. Around frame 350-400 there is some inaccuracy in the rotational DOFs as the instrument t_z value increases over this sequence.

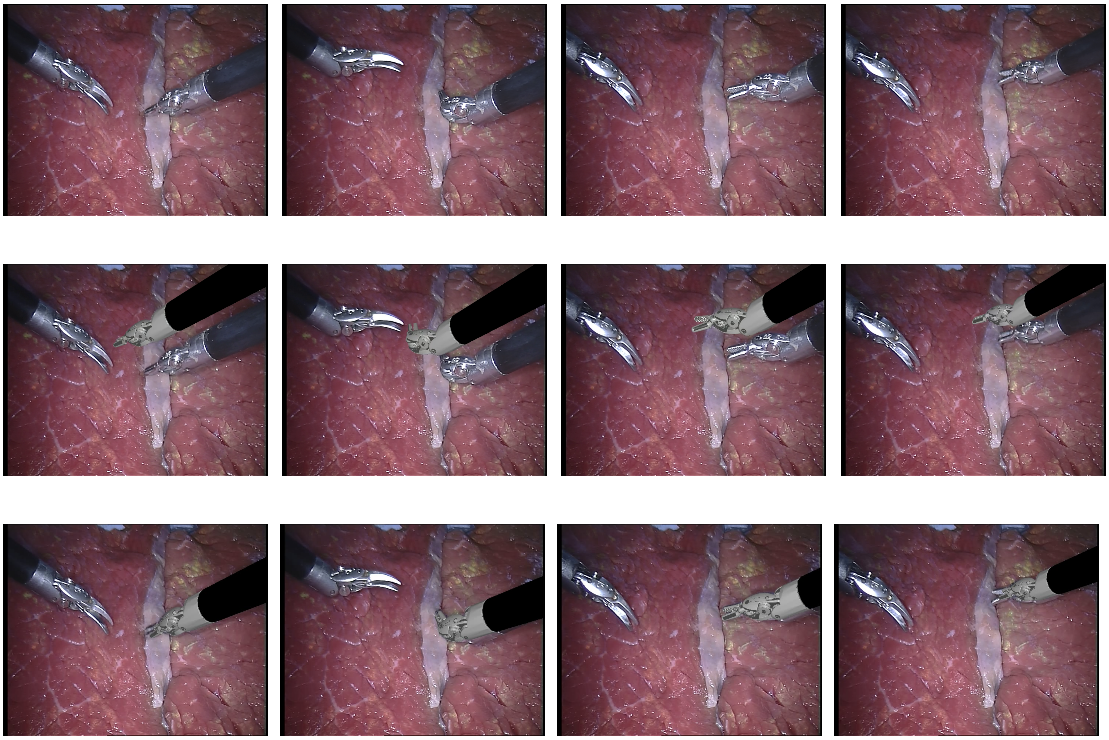


Figure 6.20: Qualitative analysis from ex-vivo dataset 4 showing frames 100, 200, 300 and 400. The top row shows the original frames, the middle row shows the output from the raw, uncorrected kinematics and the bottom row shows the MR LK tracker.

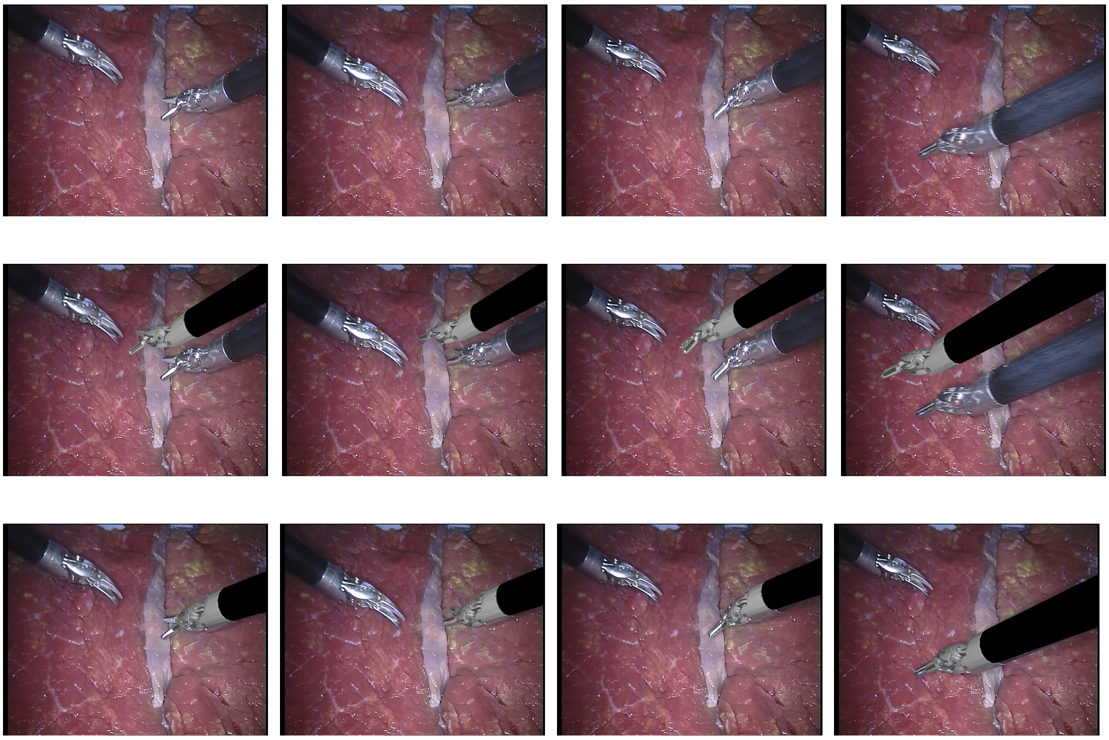


Figure 6.21: Qualitative analysis from ex-vivo dataset 4 showing frames 500, 600, 700 and 1000. The top row shows the original frames, the middle row shows the output from the raw, uncorrected kinematics and the bottom row shows the MR LK tracker.

6.6.3 Quantitative Comparison Results

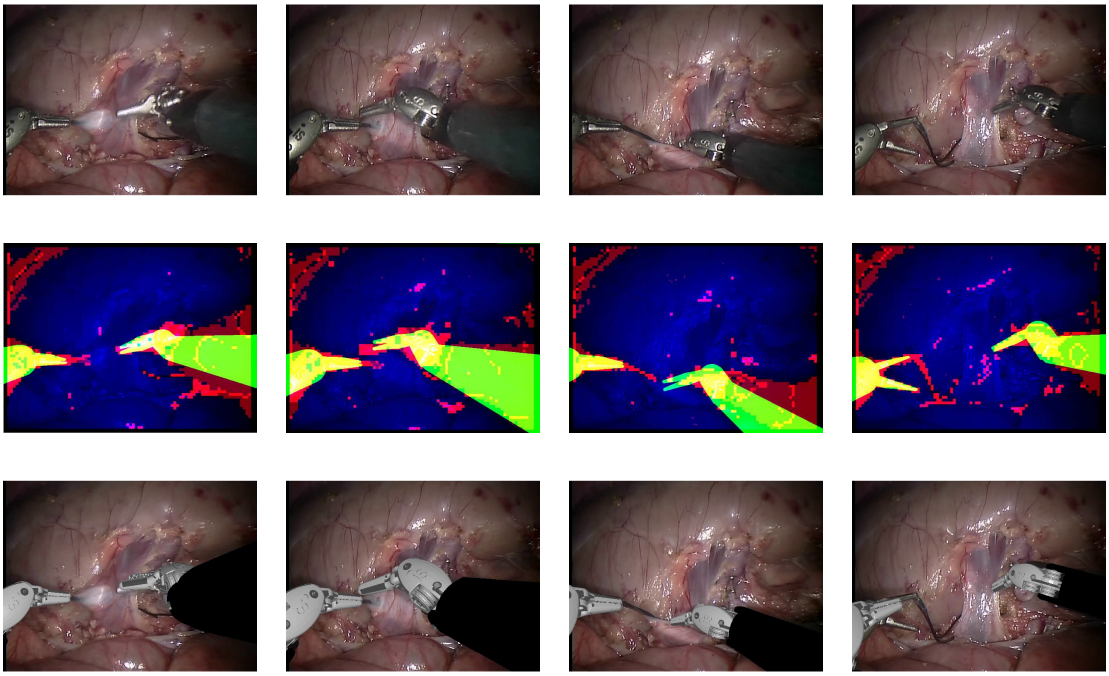


Figure 6.22: Visual comparison for the dataset of [9]. This dataset shows a challenging in-vivo sequence with 2 da Vinci LND instruments. The top row shows the raw video frames 25, 75, 125 and 175, the corresponding frames from the method of [9] are in row 2 and the frames from our method are in row 3. Although the data is challenging, both methods show good alignment. Typically our method has better alignment but the right instrument fails to track the clasper opening in frame 175, which is correctly tracked by [9].

Recent articulated robotic tracking methods [9, 10, 11] allow us to provide a quantitative comparison method between our fully visual technique and methods that combine visual tracking with robotic kinematic information. Our first comparison is between our method and that of [9] which provided a method of tracking general 3D articulated object and contained a validation section on robotic surgical instruments. This method used a similar region overlap type metric to our technique incorporating multiple instrument regions to provide added robustness. However, this was formulated within a gradient-free optimization as the simple overlap metric did not allow for analytical Jacobians to be computed. This leads to slow and often inaccurate solutions for robotic instruments although the method worked well for retinal instruments and human hands. We show results using the 4 frame evaluation used in the original paper where frames 25, 75, 125 and 175 are manually segmented. We use classification metrics of precision, recall and the F1 score to compare the overlap between the manual segmentation and the rendering of the instrument in that frame. Precision (P) and recall (R) are defined as in Equations 3.9 and 3.10 and F1 score (F1) are computed as:

$$F1 = 2\frac{P \times R}{P + R} \tag{6.8}$$

where the F1 score is the harmonic mean of the precision and recall and is often used as a weighted average of the two measures.

The original work of [9] tends to underlap the ground truth slightly, whereas our method tends to overlap slightly which is reflected in the higher precision value for [9] and the higher recall value for our work. However, when taken together, the F1 score shows much higher performance in our method. In this dataset, we make one modification to our method, as the first frame of video does not show a good view of the instrument clasper meaning the color distribution for this class was badly learned from the first frame. To counter this, we chose a later frame to learn our RF, however this is similar to the original authors who chose frames from across the video to learn their color model.

	Precision - [9]	**Recall** - [9]	**F1** - [9]	**Precision** - Ours	**Recall** - Ours	**F1** - Ours
Frame 25	**0.96**	0.70	0.81	0.84	**0.96**	**0.90**
Frame 75	**0.96**	0.85	0.90	0.83	**0.99**	**0.91**
Frame 125	0.84	0.60	0.70	**0.93**	**0.92**	**0.92**
Frame 175	**0.94**	0.80	0.87	0.90	**0.85**	0.87
Average	**0.93**	0.74	0.82	0.87	**0.93**	**0.90**

Table 6.3: Overlap precision, recall and F1 score for the 4 frames used in the evaluation in [9]. As we performed this evaluation ourselves using hand-crafted masks the results reported in this table for the method of [9] are slightly different, albeit better than the results in the original paper.

The recent method and data of [10] allows us to compare with the state-of-the-art for 3D articulated instrument tracking which combines robot kinematics with a point based detector to provide accurate real-time tracking. We evaluate on 2 phantom sequences with LND instruments which contain complex articulations which make visual tracking extremely challenging. The results are evaluated quantitative in Table 6.4 where the authors manually labeled the centre locations of several tool parts that were used in their point based detection system to obtain a ground truth. The authors then computed the relative pose between the predicted instrument location and the manually labeled instrument location for all frames in the video. Qualitative evaluation is shown in Figures 6.23 and 6.24. In our analysis of dataset 2, we encountered 1 tracking failure for our method at frame 1200 when the left instrument obtained an inaccurate pose due to a challenging period of articulation. Although both instruments go through periods of the video when they exhibit inaccurate tracking, this particular sequence was followed by a period when the instruments crossed over one another. This caused large drift in the left instrument which was deemed unrecoverable and a manual initialization was required.

Dataset	**T error (mm)** - Ours	**R error (rads)** - Ours	**T error (mm)** - [10]	**R error (rads)** - [10]
Dataset 1	5.07 ± 2.08	0.43 ± 0.26	**1.50 ± 1.12**	**0.12 ± 0.07**
Dataset 2	3.85 ± 3.64	0.58 ± 0.31	**3.14 ± 1.96**	**0.12 ± 0.08**

Table 6.4: The numerical accuracy of our method compared with [10]. The rotation and translation error is computed for each frame from the manually labeled ground truth part locations. Although our results are not as accurate as the method of [10], we are still able to obtain good tracking over the majority of the sequence and critically are not relying on kinematics to perform our estimation.

Our final comparative evaluation is with the method of [11] which also fuses kinematics with a point based detector but does not obtain real-time performance. Their evaluation was performed on 6 sequences (5 Porcine ex-vivo and 1 Porcine in-vivo) each containing 2 instruments of which we have CAD models only for the LND instrument, therefore our evaluation is limited to the 3 sequences which contain this instrument. To perform evaluation, they manually labeled the outline of the instrument in each frame of the sequence and considered a pose estimate correct if the center line of each component of the articulated model fell within the boundary of this segmented frame. We show the numerical scores for our method compared with the method of [11] in Table 6.5. In this table we also report the results of [10] where the comparison was performed in their own paper. Our results in these datasets, particularly dataset 2 and 3, are quite poor however this does not well represent that accuracy of our method as the validation metric is highly sensitive to misalignments in the clasper positions, which is where our method is weakest compared with kinematically driven pose estimation. The kinematics are very accurate at

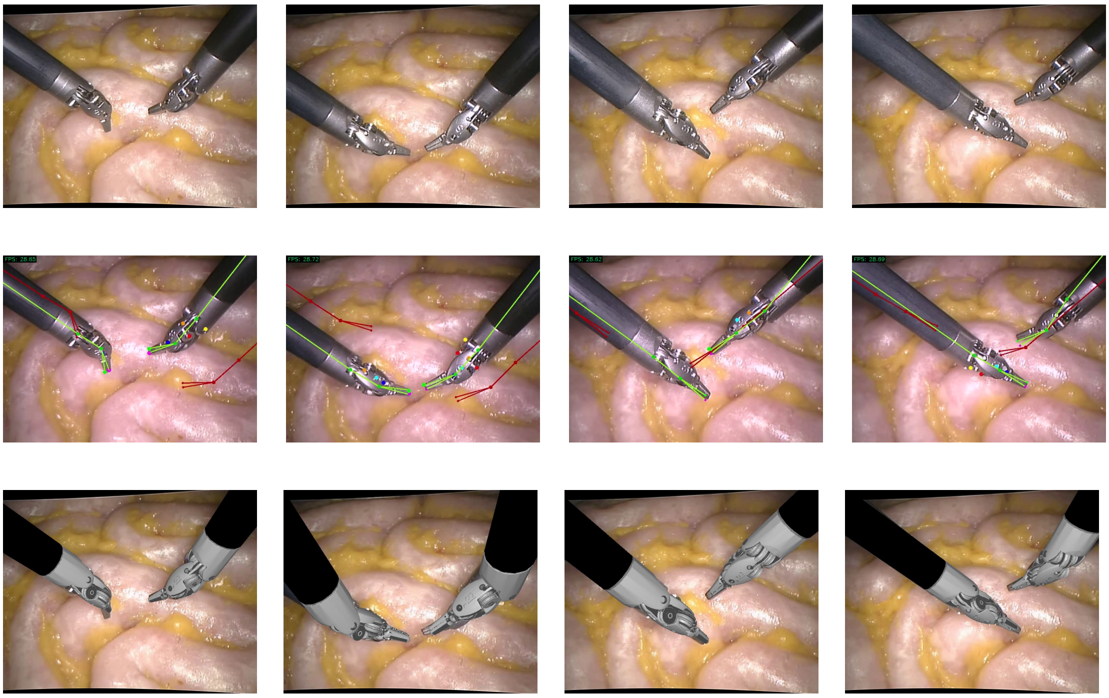


Figure 6.23: Visual comparison from dataset 1 of [10] where the top row shows the original of frames 200, 400, 750 and 950, the middle row shows the results of [10], where the green lines show their algorithm's estimate, and the bottom row shows our results. Although our method does not provide equally accurate alignment, there is still good visual overlap in most frames.

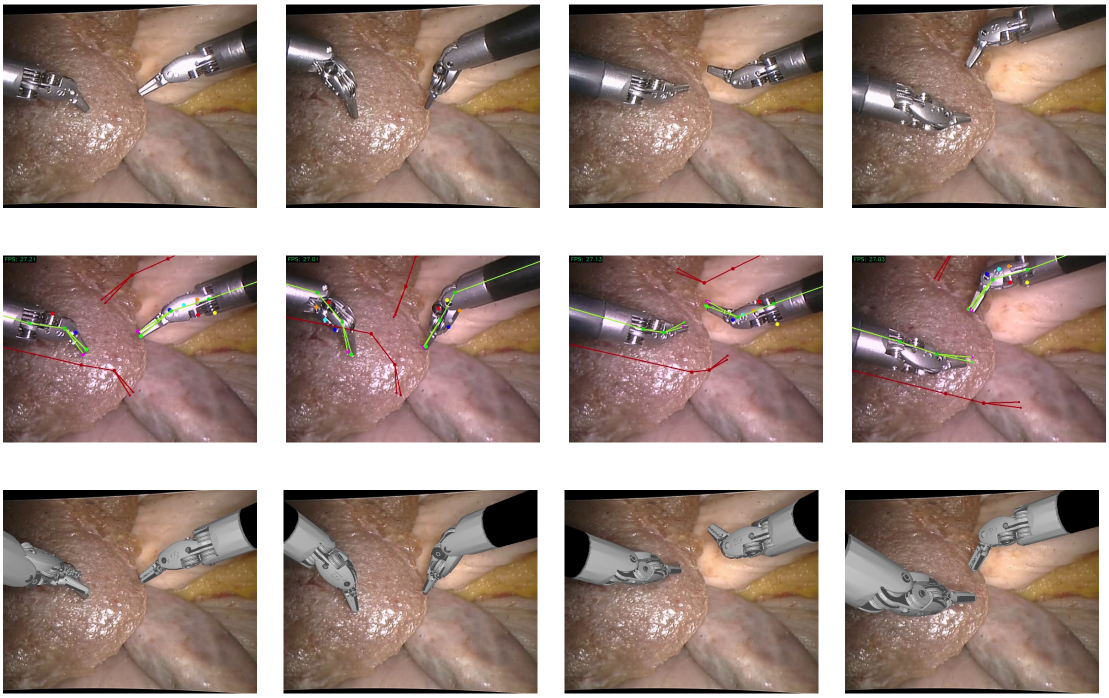


Figure 6.24: Visual comparison from dataset 2 of [10] where the top row shows the original of frames 350, 450, 900 and 1200, the middle row shows the results of [10], where the green lines show their algorithm's estimate, and the bottom row shows our results. Frame 350 shows error in the left instrument when using our method, the instrument t_z translation is clearly wrong and this prevents the wrist and clasper from reaching the correct configuration.

estimating the pose of the articulated head as these DOFs are less affected by the accumulated error in the arm. Visual methods however are most inaccurate at the end of the instrument as the clasper and head are smaller constraints compared with the shaft and small positioning errors are common. For example, in the qualitative evaluation of dataset 3 in Figure 6.27, frame 1800 is given a zero score due to tiny misalignments in the clasper rather than the shaft, whereas visually it is clear that the shaft is where most of the error lies. Additionally, frame 500 is given a zero score due to a very small misalignment at the clasper but visually the alignment appears to be good. Frame 150 gives a correct score using the overlap metric yet it visually appears much worse than other frames. Datasets 1 and 2 are recorded at 15Hz, which did not affect the original method of [11] which used tracking-by-detection. However our method reinitializes the gradient search at the estimate from the previous frame, which caused us to encounter problems when the inter-frame movement of the instrument was so large that there was no image plane overlap between the instrument in the new image and the instrument in the old image. Any region based method would fail in this case, as some overlap between the model and the data is required to obtain useful Jacobians. This situation happened 4 times in dataset 2 so required manual reinitialization at each occurrence.

Dataset	**% Correct (LND)** - Ours	**% Correct (Both)** - [11]	**% Correct (Both)** - [10]
Dataset 1	66.66	97.12	**97.79**
Dataset 2	31.57	98.04	**99.25**
Dataset 3	34.78	**98.76**	96.57

Table 6.5: The numerical accuracy of our method compared with [11] and [10]. The error metrics are computed by checking for complete overlap between a hand labeled instrument and a center line rendered version of the instrument. As the ground truth segmentations are not available for these datasets, we construct our own ground truth using the video frames. However, we segment every 50 frames, rather than every frame. Additionally, the original papers report tracking for both instruments and we give their accuracy exactly as reported in their papers. However, as we currently only have a 3D model for the LND we can only report our accuracy on this instrument.

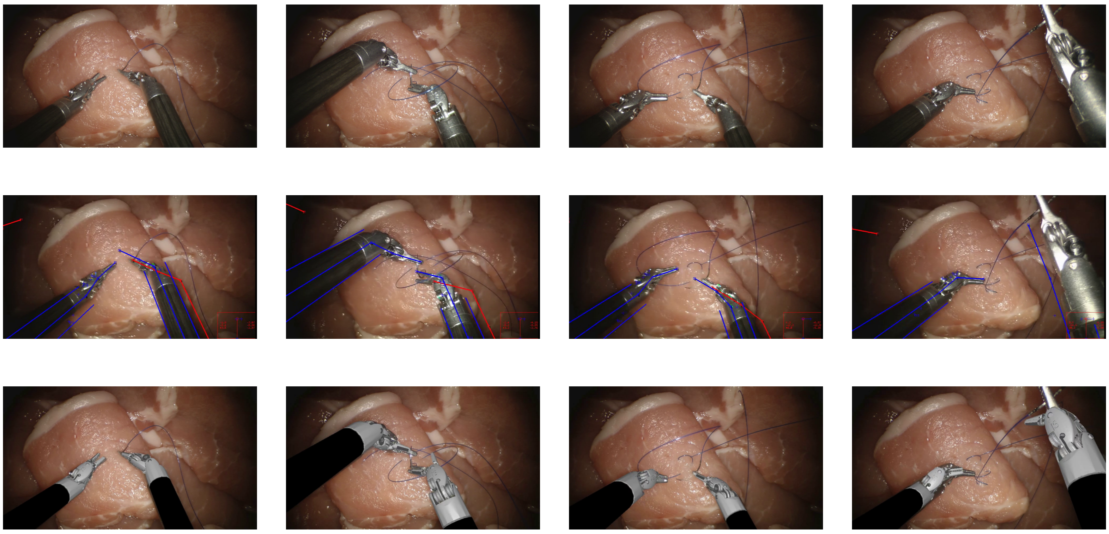


Figure 6.25: Visual comparison for the LND instrument in dataset 1 of [11]. Row 1 shows the raw frames 100, 350 and 500 and 800. The results of [11] for the equivalent frames are shown in row 2 and our results in row 3. Although the tracking is good for the majority of the sequence, clear rotational misalignment can be seen in frame 800 as the instrument moves close to the camera.

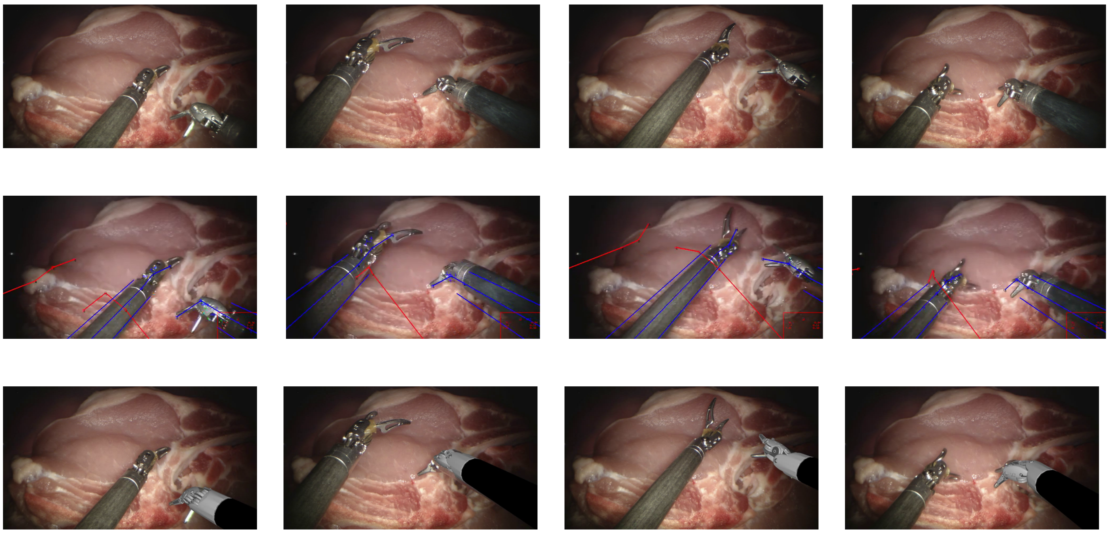


Figure 6.26: Visual comparison for the LND instrument in dataset 2 of [11]. Row 1 shows the raw frames 250, 450, 650 and 800. The results of [11] for the equivalent frames are shown in row 2 and our results in row 3. This is a complex sequence with large interframe motion which causes the tracker problems, particularly in correctly estimating the clasper opening angle. Additionally the large shadows around the edge of the image introduces complications in tracking the border between the plastic and metal on the shaft, which is nearly imperceptible in many frames.

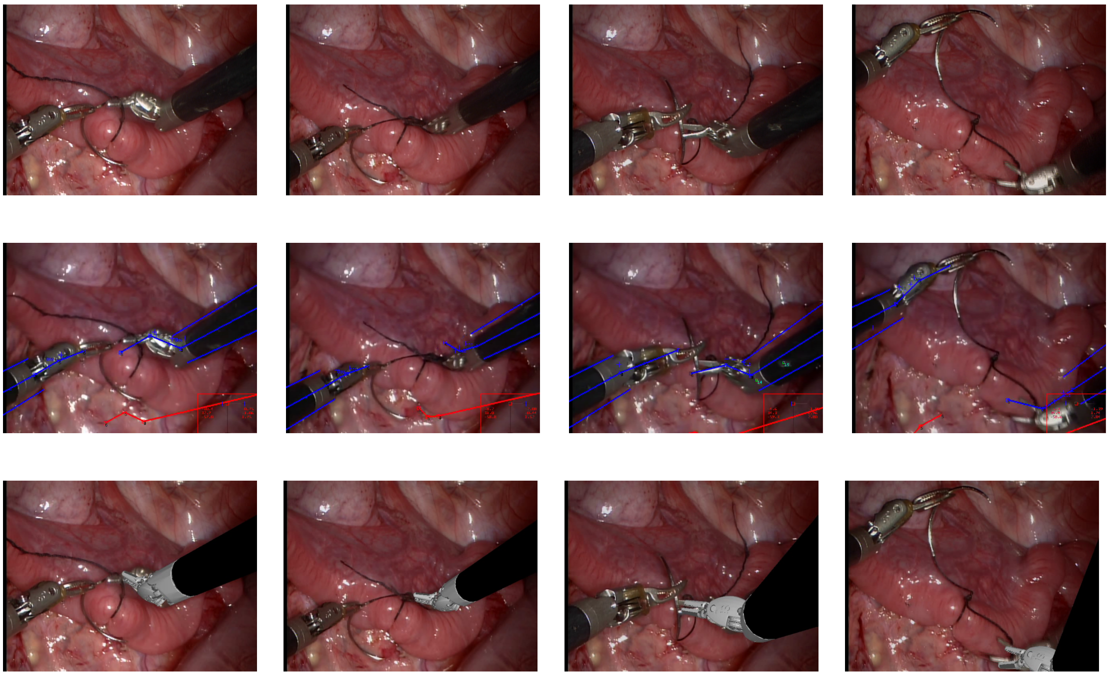


Figure 6.27: Visual comparison for the LND instrument in dataset 3 of [11]. Row 1 shows the raw frames 150, 500, 1050 and 1800. The results of [11] for the equivalent frames are shown in row 2 and our results in row 3. This sequence shows complex articulation as the instrument performs suturing yet our algorithm provides good tracking of the wrist throughout the sequence. However, the range of motion of the shaft is larger in this sequence compared with others and large errors are seen, particularly in frame 1050 when the rotation is badly misaligned.

6.7 Analysis of Performance Under Increasing Noise

Our final experiment looked at the performance of our pose estimation method under controlled levels of noise. We created a synthetic dataset by rendering an instrument model in front of a red background using kinematic data from ex-vivo dataset 2 to generate the poses for each frame. We used a Bernoulli Trial to swap the RF class probabilities where the propability of flipping is increased with each experiment (see Figure 6.28). Swapping the class probabilities involves swapping the probability assiged to the plastic shaft and the metal head and then randomly choosing one of these (with P=0.5) to swap with the background. We increased the trial probability from $\sigma = 0$ (which produces a perfect segmentation) by intervals of 0.05 until the trial probability is $\sigma = 0.8$. In Table 6.6 we show the mean and standard deviation of the error in each DOF as noise is increased and in Figures 6.29 and 6.30 we show trajectory and error plots which illustrate how the error increases as the noise is increased. The results show that for levels of noise $\sigma < 0.5$, when the image still contains at more information than noise, the tracking is quite reliable with errors within 5.38 mm for translation and 0.22 radians for rotation which is better than the ex-vivo datasets. However, at noise levels of $\sigma \geq 0.5$ tracking performance rapidly degrades with errors jumping to 50.61 mm for translation and 1.64 radians for rotation at $\sigma = 0.5$ and beyond this the trajectories have little relationship with the ground truth.

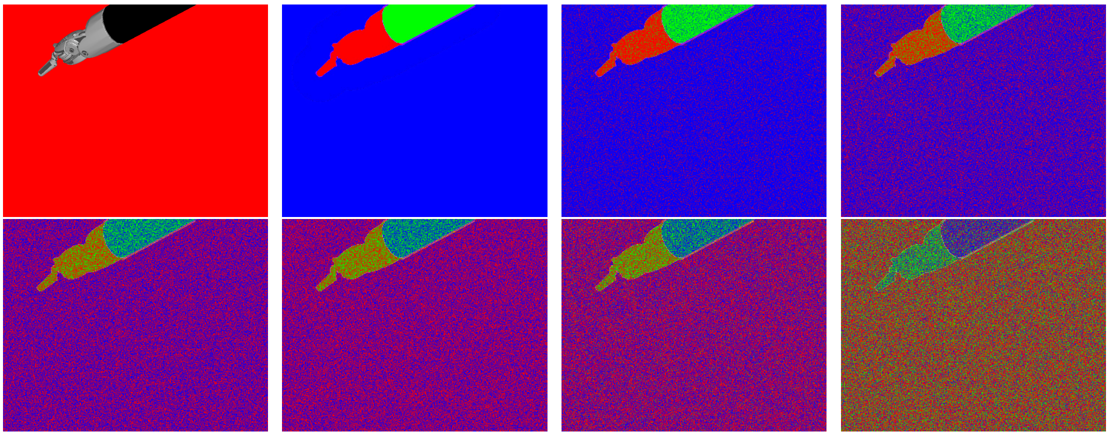


Figure 6.28: The original frame (top left) followed by the corrupted RF output for noise levels 0, 0.15, 0.3, 0.45, 0.50, 0.55 and 0.8. Despite the very limited visual change between the noise levels of $\sigma = 0.45$ and $\sigma = 0.55$ the trajectory plots in Figures 6.29 and 6.30 show a very large change in performance.

Noise Levels	$t_x\,(mm)$	$t_y\,(mm)$	$t_z\,(mm)$	$r_x\,(rads)$	$r_y\,(rads)$	$r_z\,(rads)$	wrist (rads)	grasper (rads)	grasper angle (rads)
0	0.53 ± 0.48	0.54 ± 0.50	2.44 ± 3.04	0.21 ± 0.20	0.02 ± 0.02	0.03 ± 0.03	0.05 ± 0.04	0.20 ± 0.26	0.30 ± 0.46
0.05	0.62 ± 0.56	0.55 ± 0.48	1.77 ± 1.87	0.15 ± 0.09	0.02 ± 0.01	0.02 ± 0.01	0.04 ± 0.03	0.10 ± 0.09	0.14 ± 0.18
0.1	0.62 ± 0.55	0.64 ± 0.54	2.53 ± 2.87	0.14 ± 0.11	0.02 ± 0.02	0.02 ± 0.02	0.04 ± 0.03	0.11 ± 0.11	0.16 ± 0.23
0.15	0.66 ± 0.56	0.64 ± 0.53	2.68 ± 3.29	0.14 ± 0.10	0.03 ± 0.03	0.03 ± 0.02	0.06 ± 0.06	0.10 ± 0.09	0.12 ± 0.15
0.2	0.67 ± 0.59	0.71 ± 0.55	3.23 ± 3.33	0.15 ± 0.10	0.02 ± 0.02	0.02 ± 0.02	0.04 ± 0.03	0.28 ± 0.26	0.56 ± 0.56
0.25	0.66 ± 0.57	0.70 ± 0.52	3.11 ± 2.69	0.16 ± 0.11	0.02 ± 0.02	0.02 ± 0.02	0.04 ± 0.04	0.13 ± 0.13	0.17 ± 0.23
0.3	0.77 ± 0.67	0.84 ± 0.61	3.97 ± 3.37	0.17 ± 0.11	0.02 ± 0.02	0.02 ± 0.02	0.04 ± 0.03	0.13 ± 0.15	0.25 ± 0.37
0.35	0.80 ± 0.73	0.94 ± 0.64	4.72 ± 3.66	0.16 ± 0.10	0.03 ± 0.02	0.03 ± 0.02	0.04 ± 0.04	0.14 ± 0.16	0.24 ± 0.31
0.4	0.72 ± 0.52	0.93 ± 0.53	4.28 ± 2.96	0.16 ± 0.10	0.03 ± 0.03	0.03 ± 0.03	0.06 ± 0.05	0.14 ± 0.15	0.26 ± 0.40
0.45	0.76 ± 0.58	1.02 ± 0.56	5.23 ± 3.04	0.19 ± 0.13	0.04 ± 0.04	0.03 ± 0.03	0.13 ± 0.14	0.15 ± 0.15	0.16 ± 0.17
0.5	4.39 ± 3.40	6.07 ± 3.52	50.05 ± 18.38	0.72 ± 0.42	0.43 ± 0.29	1.41 ± 0.70	0.30 ± 0.19	0.58 ± 0.38	0.90 ± 0.41
0.55	4.69 ± 4.32	11.78 ± 4.02	68.31 ± 12.15	0.25 ± 0.15	0.57 ± 0.10	0.47 ± 0.07	1.19 ± 0.20	0.87 ± 0.43	0.38 ± 0.25
0.6	5.52 ± 4.83	9.96 ± 3.87	69.11 ± 13.30	0.19 ± 0.15	0.41 ± 0.08	0.78 ± 0.10	0.54 ± 0.18	0.61 ± 0.41	1.31 ± 0.47
0.65	6.39 ± 5.27	10.17 ± 4.24	67.65 ± 12.99	0.94 ± 0.15	0.36 ± 0.05	0.54 ± 0.05	0.45 ± 0.22	0.89 ± 0.42	1.09 ± 0.22
0.7	4.90 ± 4.51	11.16 ± 3.80	68.00 ± 12.45	0.80 ± 0.20	0.58 ± 0.10	0.51 ± 0.04	0.49 ± 0.19	0.92 ± 0.41	1.59 ± 0.23
0.75	5.27 ± 4.16	8.96 ± 3.88	69.82 ± 13.10	0.36 ± 0.30	0.54 ± 0.05	0.73 ± 0.07	0.30 ± 0.18	0.92 ± 0.41	0.39 ± 0.45
0.8	5.58 ± 4.38	6.88 ± 3.93	66.99 ± 16.65	1.13 ± 0.36	0.58 ± 0.05	0.84 ± 0.10	0.26 ± 0.18	0.92 ± 0.42	1.60 ± 0.25

Table 6.6: Numerical results showing the mean error ± the standard deviation over all noise levels for all DOFs of the instrument.

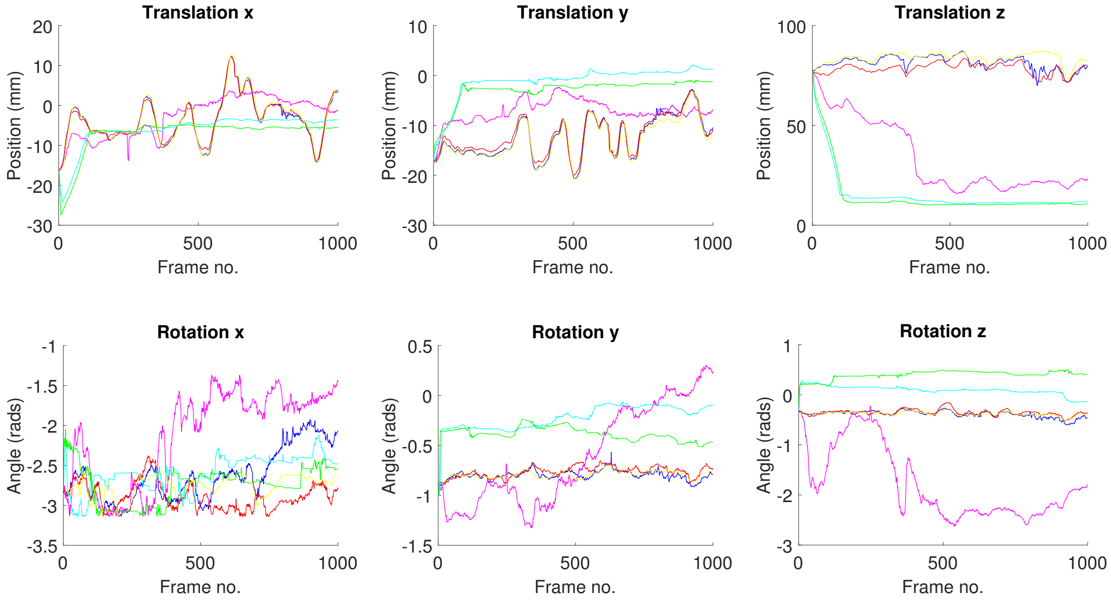


Figure 6.29: Ground Truth, $\sigma = 0.0$, $\sigma = 0.45$, $\sigma = 0.5$, $\sigma = 0.55$, $\sigma = 0.6$. Quantitative analysis of the tracking results for the rigid DOFs for the synthetic dataset with increasing noise levels. The top row shows the translation trajectories and row 2 shows the rotation trajectories where the error is low for $\sigma < 0.5$ and then rapidly increases as the noise goes beyond $\sigma = 0.5$. The trajectory at $\sigma = 0.55$, when there is more noise than information in the image shows the position of the instrument drifts to a particular configuration at frame 100 and then remains constant over the remainder of the frames.

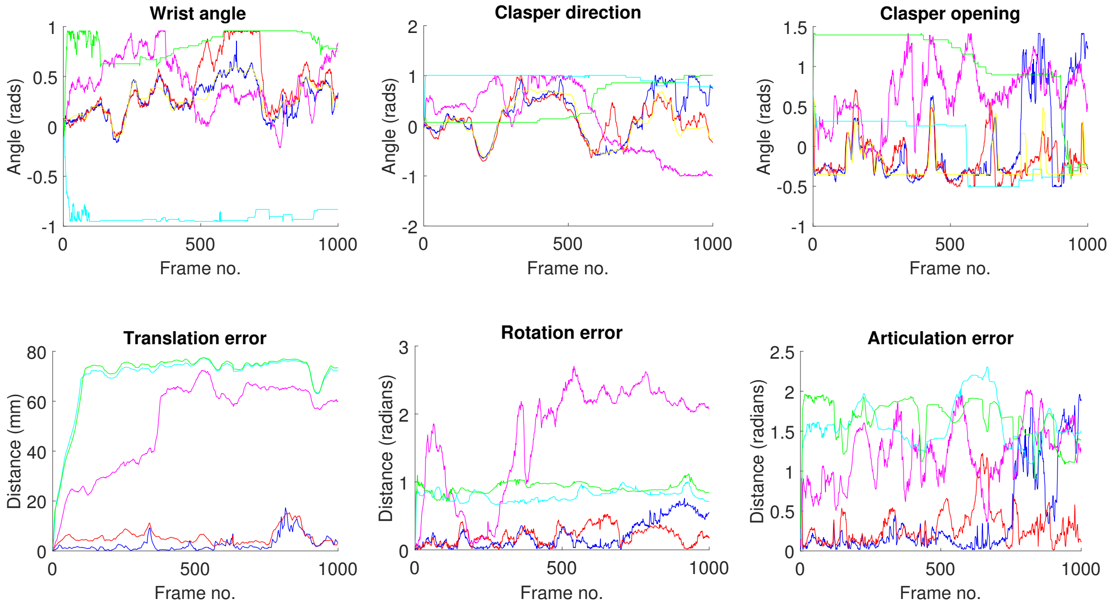


Figure 6.30: Ground Truth, $\sigma = 0.0$, $\sigma = 0.45$, $\sigma = 0.5$, $\sigma = 0.55$, $\sigma = 0.6$. Quantitative analysis of the tracking results for the articulated DOFs and the total errors for the synthetic dataset with increasing noise levels. The top row shows the articulated DOF trajectories and row 2 shows the errors for translation, rotation and articulation independently. As with the rigid DOFs, the results with $\sigma < 0.5$ follow the ground truth quite closely while the errors for $\sigma \geq 0.5$ are much larger.

6.8 Conclusion

This chapter presented a method of simultaneously tracking the 3D pose of a robotic instrument and the articulated wrist parameters by optimizing the level set segmentation and point tracking framework with kinematic based Jacobians. We demonstrate through quantitative ex-vivo datasets and quantitative and qualitative in-vivo datasets that we can accurately estimate the pose of the instrument for most sequences and accurately recover from small failures. Datasets 1, 3 and 4 show reduced accuracy in tracking the clasper compared the kinematics, with several areas of noise where the gradient descent based optimization becomes trapped in local minima which often do not resolve themselves until the true instrument configuration is altered. However, an advantage of the kinematic constraints on of the instrument means that it is impossible for the optimization to diverge too far from the true configuration. In dataset 2 we obtain excellent accuracy, tracking almost perfectly over the whole sequence due to the excellent color segmentation. When compared numerically to the kinematics, our method normally performs better when estimating the t_x and t_y degrees of freedom, with comparable accuracy in r_y and r_z whereas the kinematics are normally much better in estimating t_z, r_x and the articulated DOFs. This discrepancy is understandable due to the fact that our method is heavily driven by 2D cues, which most strongly affect the t_x, t_y and r_z DOFs whereas the kinematics is unaffected by t_z which is a significant source of error for visual methods. Additionally, the articulated DOF estimates from the kinematics are normally very accurate, which is explained by the design of the da Vinci arm (see Figure 6.3) whereby the absolute positional errors which accumulate along the arm have limited affect on these joints.

Our comparisons with [10] and [11], which both combine kinematics with visual correction using extended Kalman filters, demonstrate that this type of method remains the state-of-the-art for pure accuracy however there are still significant advantages to the pure visual methods presented in this thesis. Firstly, laparoscopic articulated instruments such as the recent FlexDex ® are unlikely to ever support access to a kinematic API and many commercial systems keep their API data private meaning methods which critically rely on this information are unable to process data collected using these systems. Further to this, requiring kinematic API access means that retrospective analysis of the many surgical videos which have been captured without kinematic information is impossible.

The synthetic experiments illustrate how accurate our method can be when the pixel classification is close to perfect and how robustly it handles Gaussian noise. However, it also shows how rapidly the performance degrades when the noise levels reach 50 $\%$ of the image information and emphasizes how critical finding reliable pixel classification methods is in producing an accurate tracker.

The major limitations that exist with our visual tracking of the articulated DOFs of the instrument center mostly on inaccurate estimation of the base frame to camera transform ${}^{cam}\mathbf{T}_{model}$. Errors in estimating this transform mean that the rotational joint axes used in the articulated DOF estimation are misaligned with the true axes, meaning that the instrument model cannot converge to alignment with the visual data. This effect can be seen in Figure 6.24 where the r_z DOF is poorly estimated meaning that the instrument head is badly misaligned. The inability of our method to recover from tracking failure, which was particularly prominent in the comparison evaluations, is another limitation of our method. To ensure that tracking can handle cases when the instrument is badly visually occluded or moves out of the camera view is essential for a tracking system that can have real in-vivo potential. As 3D generative tracking methods rely heavily on estimates from the previous frame, they are mostly unsuitable for recovery techniques as the parameter search space must be minimized to enable real-time performance. However, using 2D trackers to reinitialize 3D methods is potentially possible [176] as these methods are fast enough to perform tracking-by-detection over the whole image space.

Chapter 7

Conclusion

In this thesis, we have presented a novel method of tracking robotic and laparoscopic instruments in 3D without any required modification to the surgical workflow or the instrument design. This method has several key clinical applications such as hand-eye calibration of a surgical camera using the instruments as a calibration target [191] or in surgical skills analysis where instrument motion provides a highly significant cue for ability. As our method is not real-time at this moment, this type of retrospective analysis is the main application of our proposal however, once real-time performance is achieved applications such as depth-corrected visual-servoing of the instrument or camera become possible and direct overlay of intra-operative imaging modalities such as ultrasound provided by a pick-up probe can be provided without an attached marker [192]. Additionally, with stereo reconstruction of tissue surfaces [100], using 3D pose information we can detect interactions between the instrument and the patient's anatomy which could be an important component of simulating haptic feedback or providing virtual fixtures.

7.1 Contributions

In chapter 3 we demonstrated a robust method of labeling image pixels according to 2 instrument classes or a background class where we used numerous experiments to find the optimal setup for accurate segmentation. Finding accurate segmentation models is an open area of research in surgical instrument detection and tracking and most methods attempt to perform this without explicit shape models [79, 75] which as corroborated by our own analysis are highly sensitive to occlusion and noise in the image. In chapter 4 we demonstrate that shape can be incorporated as a strong feature within a gradient based framework to simultaneously estimate 3D pose and segment an image into instrument and tissue. Using shape in this way allows regions of the instrument which are occluded by tissue and lighting to be correctly segmented provided enough of the shape of the instrument is visible to the tracking algorithm for it to constrain the estimate. We demonstrate both quantitatively and qualitatively that we can track accurately in 3D using calibrated ex-vivo samples and in-vivo data from prostatectomy cases. There are several competing methods which can estimate the 3D pose of rigid instruments [63, 69, 62] normally using either region or gradient features. Although these methods can achieve excellent accuracy on very clean images, they are typically geometric in nature relying on cylindrical shape to fit a model to the image data. This works well for clean contours of unopened laparoscopic instruments but it provides no simple way of extension to more complex objects which are not easily represented by a simple model and additionally no validation is provided on images where parts of the instrument shape are occluded. Alternatively, our method is agnostic to the shape of the object it is tracking and additionally can incorporate model deformations which we demonstrate in chapter 6. At this moment, the proposed method is one of only 2 published 3D trackers capable of tracking the articulation of surgical instruments without significant assistance from the robot kinematics. The alternative method [9] was a general method for tracking articulated objects and was not extensively validated on robotic instruments and did not provide a gradient based method meaning the solutions are often not optimal and visual inspection shows they often exhibited significant misalignment. Our qualitative and qualitative comparison in section 6.6.3 shows that we achieve superior performance both qualitatively and quantitatively. This is likely due to the gradient based optimization which we have been able to make use of due to the level set formulation and additionally due to the motion features we have used to greatly improved our performance over region features alone.

7.2 Limitations

Although not a severe limitation, one of the most disadvantageous requirements of the proposed method is that it cannot function without an accurate 3D model of the instrument. Although these models can be constructed manually for different instrument types, the ideal case is that they are supplied by the instrument manufacturer as the measurements and precise shape are more accurate and our early experiments with simpler models demonstrated that even small modifications from the true shape can have severe effects on the accuracy of the method. Recent methods in computer vision [160] have demonstrated that it is in principle possible to reconstruct the 3D shape of objects online as they are tracked in 3D however these technique are currently limited to very clean images and some user interaction to select corresponding points between several varying views of the target object.

A second limitation of fully visual methods is dealing with situations where the instrument shaft occludes the instrument head due to the orientation of the instrument relative to the camera and specific articulations of the instrument wrist (see Figure 7.1). In this situation, the Jacobians estimated for the pose of the wrist and claspers of the instrument are not accurate and may result in the instrument moving into an articulated configuration which is far from the true solution. Although it is possible for the tracker to recover in this situation, this process is dependent on how far from the correct solution the updated estimate is. Resolving this problem is quite challenging, as noise in the image means that simple checks to estimate if the instrument wrist or claspers are very misaligned are not reliable. The most effective solution at this moment would be a manual reset of the claspers to a zero configuration once the instrument has been straightened up although this is clearly a problem that requires further research.

A significant limitation also occurs due to our use of local optical flow features which are not explicitly linked to a static object model, unlike the shape. This can lead to issues when the optical flow features incorrectly are assigned to tissue regions of the image where they can disrupt the pose estimation with no easy way of removing them. A more effective method could be to provide a more sophisticated pruning method which allows the flow features on the instrument and flow features on the tissue to be distinguished. This could be achieved online by learning the appearance of the patches used in the optical flow tracking using the model projection as a ground truth mask. This would bear some similarity to methods such as TLD [193] where tracking is used to provide labeling for a detector. Alternatively, rather than using flexible optical flow features, a more robust point detector could be used [71] which could be used to refine the sometimes inaccurate region based features. However, this may significantly increase the runtime.

A final limitation of this method is the computational runtime itself, which can, for the articulated tracking, result in processing time of ≈ 2 seconds per gradient descent step with up to 25 steps required on some sequences. However, the method that we have chosen has been shown to be highly parallelizable [110] due to the fact the cost function is summed over each pixel independently. The implementation we have produced is much less optimized with only specific functions such as the signed distance function being computed on the GPU. This had the advantage of allowing us much greater flexibility during experimentation to add and remove features without introducing bugs that would have been harder to locate in a highly optimized implementation. Re-coding the application to make full use of parallel processing is beyond the scope of this thesis but it a necessary requirement to move the method onto live studies rather than retrospective analysis.

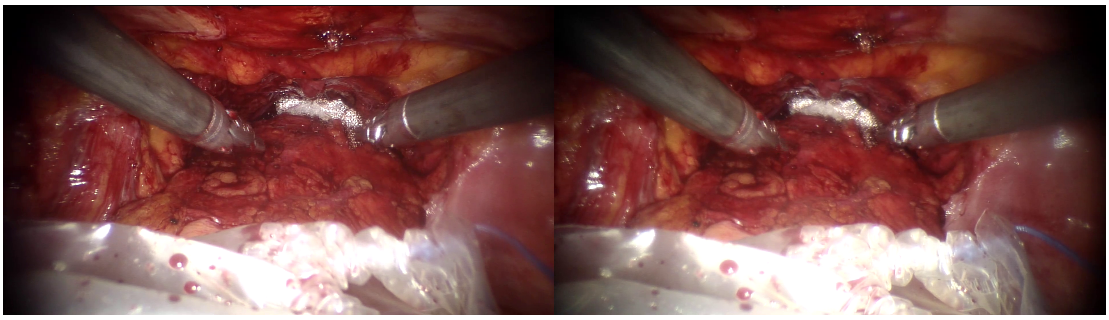


Figure 7.1: An example frame from an in-vivo prostatectomy sequence in which the articulated head of the LND instruments are positioned in such a way that the claspers and most of the head cannot be observed from the camera viewpoint. When this type of situation occurs, the results of the Jacobian update to the pose are ambiguous and may results in the claspers and head moving into a position which is far from the true location.

7.3 Future Work

Our work has numerous future extensions, the most obvious of which is to optimize the algorithm so that it can process images in real time, which we covered in the previous section. A new algorithm feature which would dramatically increase processing time on most frames would be effective convergence testing, which would prevent the algorithm from continuing to process the image while it is not improving the alignment. Our early experiments on this demonstrated that it was often not reliable over multiple datasets and entire sequences. A final optimization that would likely improve speeds considerably is to use lower resolution meshes, as much of the shape information is not contained in the detailed surface features, particularly on the instrument head which add to the computation time in the GLSL functions. A second major extension is to integrate a feature detector which links detections on surface directly back to model coordinate frame, $\mathcal{F}_{model}$. This would be similar to the existing work using kinematics [71, 10], where the kinematics based inlier detection could be replaced by the region based pose estimate. The advantage of using fixed point features and tracking-by-detection would be that the drift that is often observed when using the motion based point features would be largely eliminated.

Bibliography

[1] M. K. Gundavda and A. H. Bhandarwar, “Transoral robotic surgery in oropharyngeal carcinoma,” *Transoral Robotic Surgery in Oropharyngeal Carcinoma*, vol. 139, pp. 1389 – 1397, 2015.

[2] Y. Alassar, Y. Yildirim, S. Pecha, C. Detter, T. Deuse, and H. Reichenspurner, “Minimal access median sternotomy for aortic valve replacement in elderly patients,” *Journal of Cardiothoracic Surgery*, vol. 8, no. 1, pp. 1–5, 2013.

[3] R. G. Cohen, M. J. Mack, J. D. Finger, and L. R. J., *Minimally Invasive Cardiac Surgery*. St Louis, MO Quality Medical, 1999.

[4] A. F. Mavrogenis, O. D. Savvidou, G. Mimidis, J. Papanastasiou, D. Koulalis, N. Demertzis, and P. J. Papagelopoulos, “Computer-assisted navigation in orthopedic surgery,” *Orthopedics*, vol. 36, no. 8, pp. 631–642, 2013.

[5] H. Hirschmüller, “Stereo processing by semiglobal matching and mutual information,” *Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 328–341, 2008.

[6] G. Farnebäck, “Two-frame motion estimation based on polynomial expansion,” in *Image Analysis*, pp. 363–370, Springer, 2003.

[7] A. Reiter, P. K. Allen, and T. Zhao, “Learning features on robotic surgical tools,” in *Computer Vision and Pattern Recognition Workshops*, pp. 38–43, IEEE, 2012.

[8] W. E. L. Grimson, *Object Recognition by Computer*. The MIT Press, Cambridge, MA, 1990.

[9] Z. Pezzementi, S. Voros, and G. D. Hager, “Articulated object tracking by rendering consistent appearance parts,” in *IEEE International Conference on Robotics and Automation*, pp. 3940–3947, IEEE, 2009.

[10] M. Ye, L. Zhang, S. Giannarou, and G. Yang, “Real-time 3d tracking of articulated tools for robotic surgery,” in *Medical Image Computing and Computer-Assisted Intervention*, Springer, 2016.

[11] A. Reiter, P. K. Allen, and T. Zhao, “Appearance learning for 3d tracking of robotic surgical tools,” *The International Journal of Robotics Research*, 2013.

[12] P. Gooris, P. Worthington, and J. Evans, “Mandibulotomy: a surgical approach to oral and pharyngeal lesions,” *International journal of oral and maxillofacial surgery*, vol. 18, pp. 359–364, December 1989.

[13] A. Karl, A. Buchner, H. Beckerv, M. Staehler, M. Seitz, and C. Stief, “Perioperative blood loss in open retropubic radical prostatectomy - is it safe to get operated at an educational hospital?,” *European Journal of Medical Research*, vol. 14, no. 7, pp. 292–296, 2009.

[14] R. Agha and G. Muir, “Does laparoscopic surgery spell the end of the open surgeon?,” *Journal of the Royal Society of Medicine*, vol. 96, no. 11, pp. 544 – 546, 2003.

[15] G. Gandaglia, K. R. Ghani, and A. Sood, “Effect of minimally invasive surgery on the risk for surgical site infections: Results from the national surgical quality improvement program database,” *JAMA Surgery*, vol. 149, no. 10, pp. 1039–1044, 2014.

[16] M. M. Henry and J. N. Thompson, *Clinical Surgery*. Elsevier Health Sciences UK, 2012.

[17] J. Walsh, J. Bonnar, and F. Wright, “A study of pulmonary embolism and deep leg vien thrombosis after major gynaecological surgery using labelled fibrinogen-phlebography and lung scanning,” *BJOG: An International Journal of Obstetrics & Gynaecology*, vol. 81, no. 4, pp. 311–316, 1974.

[18] M. A. E. Ramsay, “Acute postoperative pain management,” *Proceedings (Baylor University Medical Center)*, vol. 13, no. 3, pp. 244 – 247, 2000.

[19] T. Neff, “Robotic surgery reaches a milestone, a few firsts in its wake,” *UCH Insider*, vol. 5, no. 13, 2011.

[20] T. J. Locke, T. L. Griffiths, H. Mould, and G. J. Gibson, “Rib cage mechanics after median sternotomy,” *Thorax*, vol. 5, pp. 465–468, 1990.

[21] Confederation of British Industry, “Fit for purpose, absence and workplace health survey,” tech. rep., 2013.

[22] H. G. Wakeling, M. R. McFall, C. S. Jenkins, W. G. A. Woods, W. F. A. Miles, G. R. Barclay, and S. C. Fleming, “Intraoperative oesophageal doppler guided fluid management shortens postoperative hospital stay after major bowel surgery,” *British Journal of Anaesthesia*, vol. 95, no. 5, pp. 634–642, 2005.

[23] M. J. Mack, “Minimally invasive and robotic surgery,” *The Journal of the American Medical Association*, vol. 285, pp. 568–572, Feb. 2001.

[24] A. Darzi and S. Mackay, “Recent advances in minimal access surgery,” *British Medical Journal*, vol. 324, pp. 31–34, Jan. 2002.

[25] K. H. Fuchs, “Minimally invasive surgery,” *Endoscopy*, vol. 34, no. 2, pp. 154–159, 2002.

[26] R. Pandé, O. A. Adedeji, W. M. Deanery, M. O. A. Adedeji, and F. F. Gen, “Minimally invasive surgery: Historical perspective on laparoscopic abdominal surgery,” *Historical Landmarks in Medicine*, 2014.

[27] S. D. St Peter and G. W. Holcomb II, *History of Minimally Invasive Surgery*. Elsevier, 2008.

[28] M. Ochsner, “Minimally invasive surgical procedures,” *The Ochsner Journal*, vol. 2, no. 3, p. 135136, 2000.

[29] T. Baron, “Natural orifice transluminal endoscopic surgery,” *British Journal of Surgery*, vol. 94, no. 1, p. 1, 2007.

[30] B. M. Smithers, D. C. Gotley, I. Martin, and J. M. Thomas, “Comparison of the outcomes between open and minimally invasive esophagectomy,” *Annals of Surgery*, vol. 245, no. 2, pp. 232–240, 2007.

[31] M. Buunen, R. Veldkamp, W. C. Hop, E. Kuhry, J. Jeekel, E. Haglind, L. Påhlman, M. A. Cuesta, S. Msika, M. Morino, A. Lacy, and H. Bonjer, "Survival after laparoscopic surgery versus open surgery for colon cancer: long-term outcome of a randomised clinical trial," *Lancet Oncology*, vol. 10, no. 1, pp. 44–52, 2009.

[32] The Clinical Outcomes Of Surgical Therapy Study Group, "A comparison of laparoscopically assisted and open colectomy for colon cancer," *New England Journal of Medicine*, vol. 350, no. 20, pp. 2050–2059, 2004. PMID: 15141043.

[33] M. Braga, M. Frasson, A. Vignali, W. Zuliani, V. Civelli, and V. Di Carlo, "Laparoscopic vs. open colectomy in cancer patients: long-term complications, quality of life, and survival," *Diseases of the colon and rectum*, vol. 48, no. 12, pp. 2217–23, 2005.

[34] A. Epstein, P. Groeneveld, M. Harhay, F. Yang, and D. Polsky, "Impact of minimally invasive surgery on medical spending and employee absenteeism," *JAMA Surgery*, vol. 148, no. 7, pp. 641–647, 2013.

[35] P. Miccoli, P. Berti, M. Raffaelli, G. Materazzi, S. Baldacci, and G. Rossi, "Comparison between minimally invasive video-assisted thyroidectomy and conventional thyroidectomy: A prospective randomized study," *Surgery*, vol. 130, no. 6, pp. 1039 – 1043, 2001.

[36] C. Basdogan, S. De, J. Kim, M. Muniyandi, H. Kim, and M. A. Srinivasan, "Haptics in minimally invasive surgical simulation and training," *Computer Graphics and Applications*, vol. 24, pp. 56–64, March 2004.

[37] B. M. Wolfe, Z. Szabo, M. E. Moran, P. Chan, and J. G. Hunter, "Training for minimally invasive surgery," *Surgical Endoscopy*, vol. 7, no. 2, pp. 93–95, 1993.

[38] H. Bernard and T. Hartman, "Complications after laparoscopic cholecystectomy," *American Journal of Surgery*, vol. 165, no. 4, pp. 533–535, 1993.

[39] K. Moorthy and Munz, "Motion analysis in the training and assessment of minimally invasive surgery," vol. 12, p. 137142, Jan 2003.

[40] R. H. Taylor, "A perspective on medical robotics," *Proceedings of the IEEE*, vol. 94, no. 9, pp. 1652–1664, 2006.

[41] Y. S. Kwoh, J. Hou, E. A. Jonckheere, and S. Hayati, "A robot with improved absolute positioning accuracy for ct guided stereotactic brain surgery," *Transactions on Biomedical Engineering*, vol. 35, pp. 153–160, Feb 1988.

[42] B. L. Davies, R. D. Hibberd, M. J. Coptcoat, and J. E. A. Wickham, "A surgeon robot prostatectomy - a laboratory evaluation," *Journal of Medical Engineering & Technology*, vol. 13, no. 6, pp. 273–277, 1989.

[43] W. S. Ng, B. L. Davies, R. Hibbert, A. G. Timoney, and J. E. Wickham, "A firsthand experience in transurethral resection of the prostate," *Engineering in Medicine and Biology*, pp. 120–125, 1993.

[44] A. R. Lanfranco, A. E. Castellanos, J. P. Desai, and W. C. Meyers, "Robotic surgery a current perspective," *Annals of Surgery*, vol. 239, no. 1, pp. 14–21, 2004.

[45] "Intuitive surgical annual report," tech. rep., Intuitive Surgical, Inc., 2015.

[46] D. J. Mirota, M. Ishii, and G. D. Hager, "Vision-based navigation in image-guided interventions," *Annual Review of Biomedical Engineering*, vol. 13, pp. 297–319, Aug. 2011. PMID: 21568713.

[47] T. Peters and K. Cleary, *Image-guided interventions: technology and applications*. Springer Science & Business Media, 2008.

[48] S. Park, R. Howe, and D. Torchiana, "Virtual fixtures for robotic cardiac surgery," in *Medical Image Computing and Computer-Assisted Intervention* (W. Niessen and M. Viergever, eds.), vol. 2208 of *Lecture Notes in Computer Science*, pp. 1419–1420, Springer Berlin Heidelberg, 2001.

[49] G. Moustris, S. Hiridis, K. Deliparaschos, and K. Konstantinidis, "Evolution of autonomous and semi-autonomous robotic surgical systems: a review of the literature," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 7, no. 4, pp. 375–392, 2011.

[50] S. Speidel, M. Delles, C. Gutt, and R. Dillmann, "Tracking of instruments in minimally invasive surgery for surgical skill analysis," in *International Conference on Medical Imaging and Augmented Reality*, pp. 148–155, Springer-Verlag, 2006.

[51] D. Stoyanov, "Surgical vision," *Annals of Biomedical Engineering*, vol. 40, no. 2, pp. 332–345, 2012.

[52] M. K. Chmarra, C. A. Grimbergen, and J. Dankelman, "Systems for tracking minimally invasive surgical instruments," *Minimally Invasive Therapy & Allied Technologies*, vol. 16, no. 6, pp. 328–340, 2007.

[53] S. Speidel, G. Sudra, J. Senemaud, M. Drentschew, B. P. Müller-Stich, C. Gutt, and R. Dillmann, "Recognition of risk situations based on endoscopic instrument tracking and knowledge based situation modeling," in *Medical Imaging 2008: Visualization, Image-Guided Procedures, and Modeling*, vol. 6918, 2008.

[54] D. Bouget, M. Allan, D. Stoyanov, and P. Jannin, "Vision-based and marker-less surgical tool detection and tracking: a review of the literature," *Medical Image Analysis*, vol. 35, pp. 633–654, 2017.

[55] C. Lee, Y. Wang, D. Uecker, and Y. Wang, "Image analysis for automated tracking in robot-assisted endoscopic surgery," in *International Conference on Pattern Recognition*, vol. 1, pp. 88 –92 vol.1, Oct. 1994.

[56] G. Wei, K. Arbter, and G. Hirzinger, "Real-time visual servoing for laparoscopic surgery. controlling robot motion with color image segmentation," *Engineering in Medicine and Biology Magazine*, vol. 16, pp. 40 –45, Feb. 1997.

[57] B. P. L. Lo, A. Darzi, and G.-Z. Yang, "Episode classification for the analysis of Tissue/Instrument interaction with multiple visual cues," *Medical Image Computing and Computer-Assisted Intervention*, vol. 2878, pp. 230–237, 2003.

[58] C. Doignon, F. Nageotte, and M. De Mathelin, "Detection of grey regions in color images : application to the segmentation of a surgical instrument in robotized laparoscopy," in *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 4, pp. 3394 – 3399 vol.4, Oct. 2004.

[59] D. Burschka, J. J. Corso, M. Dewan, W. Lau, M. Li, H. Lin, P. Marayong, N. Ramey, G. D. Hager, B. Hoffman, D. Larkin, and C. Hasser, "Navigating inner space: 3-D assistance for minimally invasive surgery," in *Workshop Advances in Robot Vision in conjunction with the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 67–78, 2004.

[60] S. McKenna, H. N. Charif, and T. Frank, "Towards video understanding of laparoscopic surgery: Instrument tracking," in *Proceedings of Image and Vision Computing New Zealand*, 2005.

[61] C. Doignon, P. Graebling, and M. d. Mathelin, "Real-time segmentation of surgical instruments inside the abdominal cavity using a joint hue saturation color feature," *Real-Time Imaging*, vol. 11, no. 5-6, pp. 429–442, 2005.

[62] O. Tonet, T. U. Ramesh, G. Megali, and P. Dario, "Tracking endoscopic instruments without localizer: image analysis-based approach," *Studies in Health Technology and Informatics*, vol. 119, pp. 544–549, 2006.

[63] S. Voros, J. Long, and P. Cinquin, "Automatic detection of instruments in laparoscopic images: A first step towards high-level command of robotic endoscopic holders," *The International Journal of Robotics Research*, vol. 26, pp. 1173 –1190, Nov. 2007.

[64] C. Doignon and M. de Mathelin, "A degenerate Conic-Based method for a direct fitting and 3-D pose of cylinders with a single perspective view," in *Robotics and Automation, 2007 IEEE International Conference on*, pp. 4220 –4225, Apr. 2007.

[65] A. M. Cano, F. Gayá, P. Lamata, P. Sánchez-González, and E. J. Gómez, "Laparoscopic tool tracking method for augmented reality surgical applications," in *Biomedical Simulation*, pp. 191–196, Springer, 2008.

[66] S. Speidel, J. Benzeko, S. Krappe, G. Sudra, P. Azad, B. P. Muller-Stich, C. Gutt, and R. Dillmann, "Automatic classification of minimally invasive instruments based on endoscopic image sequences," in *In Proceedings of SPIE*, vol. 7261, 2009.

[67] R. Richa, M. Balicki, E. Meisner, R. Sznitman, R. Taylor, and G. Hager, "Visual tracking of surgical tools for proximity detection in retinal surgery," in *Information Processing in Computer Assisted Interventions*, IPCAI'11, pp. 55–66, 2011.

[68] R. Sznitman, A. Basu, R. Richa, J. Handa, P. Gehlbach, R. Taylor, B. Jedynak, and G. Hager, "Unified detection and tracking in retinal microsurgery," in *Medical Image Computing and Computer-Assisted Intervention* (G. Fichtinger, A. Martel, and T. Peters, eds.), vol. 6891 of *Lecture Notes in Computer Science*, pp. 1–8, 2011.

[69] R. Wolf, J. Duchateau, P. Cinquin, and S. Voros, "3D tracking of laparoscopic instruments using statistical and geometric modeling," *Medical Image Computing and Computer-Assisted Intervention*, vol. 14, no. Pt 1, pp. 203–210, 2011. PMID: 22003618.

[70] A. Reiter, P. K. Allen, and T. Zhao, "Feature classification for tracking articulated surgical tools," in *Medical Image Computing and Computer-Assisted Intervention*, pp. 592–600, Springer, 2012.

[71] R. Austin, A. P. K, and Z. Tao, "Marker-less articulated surgical tool detection," in *Computer Assisted Radiology and Surgery*, 2012.

[72] R. Sznitman, K. Ali, R. Richa, R. Taylor, G. Hager, and P. Fua, "Data-driven visual tracking in retinal microsurgery," in *Medical Image Computing and Computer-Assisted Intervention*, vol. 7511, pp. 568–575, 2012.

[73] R. Richa, M. Balicki, R. Sznitman, E. Meisner, R. Taylor, and G. Hager, "Vision-based proximity detection in retinal surgery," *Transactions on Biomedical Engineering*, vol. 59, no. 8, pp. 2291–2301, 2012.

[74] S. Kumar, M. S. Narayanan, P. Singhal, J. J. Corso, and V. Krovi, "Product of tracking experts for visual tracking of surgical tools," in *International Conference on Automation Science and Engineering*, pp. 480–485, IEEE, 2013.

[75] S. Speidel, E. Kuhn, S. Bodenstedt, S. Röhl, H. Kenngott, B. Müller-Stich, and R. Dillmann, "Visual tracking of da vinci instruments for laparoscopic surgery," in *SPIE Medical Imaging*, pp. 903608–903608, International Society for Optics and Photonics, 2014.

[76] Y. Li, C. Chen, X. Huang, and J. Huang, "Instrument tracking via online learning in retinal microsurgery," in *Medical Image Computing and Computer-Assisted Intervention*, pp. 464–471, Springer, 2014.

[77] R. Sznitman, C. Becker, and P. Fua, "Fast part-based classification for instrument detection in minimally invasive surgery," in *Medical Image Computing and Computer-Assisted Intervention*, pp. 692–699, Springer, 2014.

[78] J. Zhou and S. Payandeh, "Visual tracking of laparoscopic instruments," *Journal of Automation and Control Engineering Vol*, vol. 2, no. 3, 2014.

[79] S. Bodenstedt, M. Wagner, B. Mayer, K. Stemmer, H. Kenngott, B. Müller-Stich, R. Dillmann, and S. Speidel, "Image-based laparoscopic bowel measurement," *International Journal of Computer Assisted Radiology and Surgery*, vol. 11, no. 3, pp. 407–419, 2015.

[80] M. Alsheakhali, M. Yigitsoy, A. Eslami, and N. Navab, "Surgical tool detection and tracking in retinal microsurgery," in *SPIE Medical Imaging*, pp. 941511–941511, International Society for Optics and Photonics, 2015.

[81] N. Rieke, D. Tan, M. Alsheakhali, F. Tombari, C. di San Filippo, V. Belagiannis, A. Eslami, and N. Navab, "Surgical tool tracking and pose estimation in retinal microsurgery," in *Medical Image Computing and Computer-Assisted Intervention* (N. Navab, J. Hornegger, W. Wells, and A. Frangi, eds.), vol. 9349 of *Lecture Notes in Computer Science*, pp. 266–273, 2015.

[82] D. Bouget, R. Benenson, M. Omran, L. Riffaud, B. Schiele, and P. Jannin, "Detecting surgical tools by modelling local appearance and global shape," *Transactions on Medical Imaging*, vol. PP, no. 99, pp. 1–1, 2015.

[83] S. Bodenstedt, J. Grtler, M. Wagner, H. Kenngott, B. P. Mller-Stich, R. Dillmann, and S. Speidel, "Superpixel-based structure classification for laparoscopic surgery," vol. 9786, pp. 978618–978618–6, 2016.

[84] L. Luccheseyz and S. Mitray, "Color image segmentation: A state-of-the-art survey," *Proceedings of the Indian National Science Academy (INSA-A)*, vol. 67, no. 2, pp. 207–221, 2001.

[85] C. Bibby and I. Reid, "Robust Real-Time visual tracking using Pixel-Wise posteriors," in *Proceedings of the 10th European Conference on Computer Vision*, pp. 831–844, 2008.

[86] G. Wyszecki and W. S. Stiles, *Color Science: Concepts and Methods, Quantitative Data and Formulae*. Wiley, 1982.

[87] M. Tkalcic, J. F. Tasic, *et al.*, “Colour spaces: perceptual, historical and applicational background,” in *Eurocon*, 2003.

[88] N. Otsu, “A threshold selection method from gray-level histograms,” *Transactions on Systems, Man and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[89] A. dos Anjos and H. Shahbazkia, “Bi-level image thresholding - A fast method,” in *International Conference on Biomedical Electronics and Devices*, pp. 70–76, 2008.

[90] G. W. Meyer and D. P. Greenberg, “Perceptual color spaces for computer graphics,” in *Proceedings of the 7th annual conference on Computer graphics and interactive techniques*, p. 254261, ACM, 1980.

[91] T. Gevers and H. Stokman, “Classifying color edges in video into shadow-geometry, highlight, or material transitions,” *Transactions on Multimedia*, vol. 5, pp. 237 – 243, June 2003.

[92] S. Haase, J. Wasza, T. Kilgus, and J. Hornegger, “Laparoscopic instrument localization using a 3-d time-of-flight/rgb endoscope,” in *Workshop on Applications of Computer Vision*, pp. 449–454, IEEE, 2013.

[93] R. Haralick, “Statistical and structural approaches to texture,” *Proceedings of the IEEE*, vol. 67, no. 5, pp. 786–804, 1979.

[94] J. Malik, S. Belongie, T. Leung, and J. Shi, “Contour and texture analysis for image segmentation,” *International Journal of Computer Vision*, vol. 43, no. 1, pp. 7–27, 2001.

[95] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Computer Vision and Pattern Recognition*, vol. 1, pp. 886 –893 vol. 1, June 2005.

[96] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained part based models,” *Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[97] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Computer Vision and Pattern Recognition*, pp. 580–587, 2014.

[98] D. Lowe, “Object recognition from local scale-invariant features,” in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 2, pp. 1150 –1157 vol.2, 1999.

[99] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, “Speeded-Up robust features (SURF),” *Proceedings of Computer Vision and Image Understanding*, vol. 110, pp. 346–359, June 2008.

[100] D. Stoyanov, M. V. Scarzanella, P. Pratt, and G.-Z. Yang, “Real-time stereo reconstruction in robotically assisted minimally invasive surgery,” *Medical Image Computing and Computer-Assisted Intervention*, vol. 13, no. Pt 1, pp. 275–282, 2010. PMID: 20879241.

[101] T. Schoenemann and D. Cremers, “Near real-time motion segmentation using graph cuts,” in *Pattern Recognition*, pp. 455–464, Springer, 2006.

[102] D. Comaniciu and P. Meer, “Mean shift: a robust approach toward feature space analysis,” *Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 603–619, May 2002.

[103] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[104] V. Lepetit and P. Fua, “Keypoint recognition using randomized trees,” *Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 1465 –1479, Sept. 2006.

[105] P. Kontschieder, S. Bulo, H. Bischof, and M. Pelillo, “Structured class-labels in random forests for semantic image labelling,” in *International Conference on Computer Vision*, pp. 2190 –2197, Nov. 2011.

[106] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, pp. 5–32, Oct. 2001.

[107] A. Y. Ng and M. I. Jordan, “On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes,” in *Advances in Neural Information Processing Systems*, pp. 841–848, 2002.

[108] S. Prince, *Computer Vision: Models Learning and Inference*. Cambridge University Press, 2012.

[109] D. Cremers, “Dynamical statistical shape priors for level set-based tracking,” *Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 8, pp. 1262–1273, 2006.

[110] V. A. Prisacariu and I. D. Reid, “PWP3D: Real-Time segmentation and tracking of 3D objects,” *Interational Journal of Computer Vision*, vol. 98, pp. 335–354, Jan. 2012.

[111] T. Drummond and R. Cipolla, “Real-time visual tracking of complex structures,” *Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 932 –946, July 2002.

[112] V. Lepetit, F. Moreno-Noguer, and P. Fua, “Epnp: An accurate o(n) solution to the pnp problem,” *International Journal of Computer Vision*, vol. 81, no. 2, pp. 155–166, 2009.

[113] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second ed., 2004.

[114] F. Moreno-Noguer, V. Lepetit, and P. Fua, “Accurate non-iterative o(n) solution to the pnp problem,” in *International Conference on Computer Vision*, pp. 1–8, 2007.

[115] W. Zhao, C. J. Hasser, W. C. Nowlin, and B. D. Hoffman, “Methods and systems for robotic instrument tool tracking with adaptive fusion of kinematics information and image information,” 2012. US Patent 8,108,072.

[116] S. Hinterstoisser, S. Holzer, C. Cagniart, S. Ilic, K. Konolige, N. Navab, and V. Lepetit, “Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes,” in *International Conference on Computer Vision*, pp. 858–865, Nov 2011.

[117] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, “Real-time human pose recognition in parts from single depth images,” in *Computer Vision and Pattern Recognition*, pp. 1297–1304, 2011.

[118] G. Gkioxari, P. Arbelaez, L. Bourdev, and J. Malik, “Articulated pose estimation using discriminative armlet classifiers,” in *Computer Vision and Pattern Recognition*, pp. 3342–3349, 2013.

[119] M. Yamada, L. Sigal, and M. Raptis, "Covariate shift adaptation for discriminative 3d pose estimation," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, pp. 235–247, Feb 2014.

[120] P. Felzenszwalb, R. Girshick, and D. McAllester, "Cascade object detection with deformable part models," in *Computer Vision and Pattern Recognition*, 2010.

[121] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[122] M. Isard and A. Blake, "CONDENSATION - conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29, pp. 5–28, 1998.

[123] X. Ren and J. Malik, "Tracking as repeated figure/ground segmentation," in *Computer Vision and Pattern Recognition*, pp. 1–8, IEEE, 2007.

[124] J. Shotton, M. Johnson, and R. Cipolla, "Semantic texton forests for image categorization and segmentation," in *Computer Vision and Pattern Recognition*, pp. 1–8, IEEE, 2008.

[125] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Computer Vision and Pattern Recognition*, pp. 3431–3440, 2015.

[126] M. Allan, S. Ourselin, S. Thompson, D. J. Hawkes, J. Kelly, and D. Stoyanov, "Toward detection and localization of instruments in minimally invasive surgery," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 4, pp. 1050–1058, 2013.

[127] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr, "Conditional random fields as recurrent neural networks," in *International Conference on Computer Vision*, pp. 1529–1537, 2015.

[128] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[129] G. Bradski *Dr. Dobb's Journal of Software Tools*, 2000.

[130] A. Criminisi, "Decision forests: A unified framework for classification, regression, density estimation, manifold learning and Semi-Supervised learning," *Foundations and Trends in Computer Graphics and Vision*, vol. 7, no. 2-3, pp. 81–227, 2011.

[131] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, no. 1, pp. 379–423, 1948.

[132] G. Louppe, L. Wehenkel, A. Sutera, and P. Geurts, "Understanding variable importances in forests of randomized trees," in *Advances in Neural Information Processing Systems*, pp. 431–439, 2013.

[133] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297.

[134] P. M. Granitto, C. Furlanello, F. Biasioli, and F. Gasperi, "Recursive feature elimination with random forest for ptr-ms analysis of agroindustrial products," *Chemometrics and Intelligent Laboratory Systems*, vol. 83, no. 2, pp. 83 – 90, 2006.

[135] B. H. Menze, B. M. Kelm, R. Masuch, U. Himmelreich, P. Bachert, W. Petrich, and F. A. Hamprecht, "A comparison of random forest and its gini importance with standard chemometric methods for the feature selection and classification of spectral data," *BMC Bioinformatics*, vol. 10, no. 1, pp. 1–16, 2009.

[136] R. Baeza-Yates, B. Ribeiro-Neto, *et al.*, *Modern information retrieval*, vol. 463. ACM press New York, 1999.

[137] L. Maier-Hein, S. Mersmann, D. Kondermann, S. Bodenstedt, A. Sanchez, C. Stock, H. G. Kenngott, M. Eisenmann, and S. Speidel, "Can masses of non-experts train highly accurate image classifiers?," in *Medical Image Computing and Computer-Assisted Intervention*, pp. 438–445, Springer, 2014.

[138] Z. Zhang, "A flexible new technique for camera calibration," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[139] V. Lepetit and P. Fua, "Monocular Model-Based 3D tracking of rigid objects: A survey," in *Foundations and Trends in Computer Graphics and Vision*, pp. 1–89, 2005.

[140] T. Brox, B. Rosenhahn, J. Gall, and D. Cremers, "Combined region and motion-based 3d tracking of rigid and articulated objects," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 3, pp. 402–415, 2010.

[141] M. Allan, S. Thompson, M. J. Clarkson, S. Ourselin, D. J. Hawkes, J. Kelly, and D. Stoyanov, "2d-3d pose tracking of rigid instruments in minimally invasive surgery," in *Information Processing in Computer-Assisted Interventions*, vol. 8498, pp. 1–10, 2014.

[142] M. Allan, P.-L. Chang, S. Ourselin, D. J. Hawkes, A. Sridhar, J. Kelly, and D. Stoyanov, "Image based surgical instrument pose estimation with multi-class labelling and optical flow," in *Medical Image Computing and Computer-Assisted Intervention*, pp. 331–338, Springer, 2015.

[143] V. A. Prisacariu and I. Reid, "Nonlinear shape manifolds as shape priors in level set segmentation and tracking," in *Computer Vision and Pattern Recognition*, pp. 2185–2192, IEEE Computer Society, 2011.

[144] L. Bar, T. F. Chan, G. Chung, M. Jung, N. Kiryati, R. Mohieddine, N. Sochen, and L. A. Vese, "Mumford and shah model and its applications to image segmentation andimage restoration," in *Handbook of mathematical methods in imaging*, pp. 1095–1157, Springer, 2011.

[145] D. Mumford and J. Shah, "Optimal approximations by piecewise smooth functions and associated variational problems," *Communications on Pure and Applied Mathematics*, vol. 42, no. 5, pp. 577–685, 1989.

[146] T. Pock, D. Cremers, H. Bischof, and A. Chambolle, "An algorithm for minimizing the mumford-shah functional," in *International Conference on Computer Vision*, pp. 1133–1140, IEEE, 2009.

[147] J. Gall, B. Rosenhahn, and H.-P. Seidel, "Robust pose estimation with 3d textured models," in *Advances in Image and Video Technology*, pp. 84–95, Springer, 2006.

[148] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[149] J. Shi and J. Malik, “Normalized cuts and image segmentation,” *Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[150] J. Wang, *Graph Based Image Segmentation: A Modern Approach.* VDM Verlag, 2008.

[151] D. Cremers, M. Rousson, and R. Deriche, “A review of statistical approaches to level set segmentation: Integrating color, texture, motion and shape,” *Interational Journal of Computer Vision*, vol. 72, pp. 195–215, Apr. 2007.

[152] T. Chan and L. Vese, “Active contours without edges,” *Transactions on Image Processing*, vol. 10, pp. 266 –277, Feb. 2001.

[153] M. Kass, A. Witkin, and D. Terzopoulos, “Snakes: Active contour models,” *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.

[154] C. Epstein and M. Gage, “The curve shortening flow,” in *Wave motion: theory, modelling, and computation*, pp. 15–59, Springer, 1987.

[155] N. Paragios, O. Mellina-Gottardo, and V. Ramesh, “Gradient vector flow fast geodesic active contours,” in *International Conference on Computer Vision*, vol. 1, pp. 67–73, IEEE, 2001.

[156] S. Osher and J. A. Sethian, “Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations,” *Journal of Computational Physics*, vol. 79, pp. 12–49, Nov. 1988.

[157] C. Samson, L. Blanc-Féraud, G. Aubert, and J. Zerubia, “A level set model for image classification,” *International Journal of Computer Vision*, vol. 40, no. 3, pp. 187–197, 2000.

[158] S. Dambreville, R. Sandhu, A. Yezzi, and A. Tannenbaum, “Robust 3D pose estimation and efficient 2D region-based segmentation from a 3D shape prior,” in *European Conference on Computer Vision*, pp. 169–182, 2008.

[159] B. Rosenhahn, T. Brox, and J. Weickert, “Three-dimensional shape knowledge for joint image segmentation and pose estimation,” in *Pattern Recognition*, p. 109116, Springer, 2005.

[160] T. Cashman and A. Fitzgibbon, “What shape are dolphins? building 3D morphable models from 2D images,” *Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 232–244, 2013.

[161] C. Schmaltz, B. Rosenhahn, T. Brox, D. Cremers, L. Wietzke, and G. Sommer, “Region-based pose tracking,” in *In Proc. 3rd Iberian Conference on Pattern Recognition and Image Analysis*, 2007.

[162] T. Chan and W. Zhu, “Level set based shape prior segmentation,” in *Computer Vision and Pattern Recognition*, vol. 2, pp. 1164–1170, IEEE, 2005.

[163] C. Schmaltz, B. Rosenhahn, T. Brox, and J. Weickert, “Region-based pose tracking with occlusions using 3d models,” *Machine vision and applications*, vol. 23, no. 3, pp. 557–577, 2012.

[164] W. R. Hamilton, *Elements of quaternions.* Longmans, Green, & Company, 1866.

[165] M. D. Wheeler and K. Ikeuchi, *Iterative estimation of rotation and translation using the quaternion.* Carnegie-Mellon University. Department of Computer Science, 1995.

[166] K. Shoemake, “Animating rotation with quaternion curves,” in *ACM SIGGRAPH*, vol. 19, pp. 245–254, ACM, 1985.

[167] P. F. Felzenszwalb and D. P. Huttenlocher, “Distance transforms of sampled functions,” tech. rep., Cornell Computing and Information Science, 2004.

[168] P. Kazanzides, Z. Chen, A. Deguet, G. Fischer, R. Taylor, and S. DiMaio, “An open-source research kit for the da vinci surgical system,” in *International Conference on Robotics and Automation*, IEEE, 2014.

[169] S. DiMaio and C. Hasser, “The da vinci research interface,” in *MICCAI Workshop on Systems and Archicture for Computer Assisted Interventions, Midas Journal*, 07 2008.

[170] E. W. Weisstein, “Euler angles.” `http://mathworld.wolfram.com/EulerAngles.html`. Accessed: 2016-06-29.

[171] V. Lepetit, J. Pilet, and P. Fua, “Point matching as a classification problem for fast and robust object pose estimation,” in *Computer Vision and Pattern Recognition*, vol. 2, pp. II–244, 2004.

[172] G. Klein and D. Murray, “Parallel tracking and mapping for small ar workspaces,” in *International Symposium on Mixed and Augmented Reality*, pp. 225–234, IEEE, 2007.

[173] O. Faugeras, *Three-dimensional computer vision: a geometric viewpoint*. MIT press, 1993.

[174] C.-P. Lu, G. D. Hager, and E. Mjolsness, “Fast and globally convergent pose estimation from video images,” *Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 6, pp. 610–622, 2000.

[175] D. Oberkampf, D. F. DeMenthon, and L. S. Davis, “Iterative pose estimation using coplanar points,” in *Computer Vision and Pattern Recognition*, pp. 626–627, 1993.

[176] X. Du, M. Allan, A. Dore, S. Ourselin, D. Hawkes, J. D. Kelly, and D. Stoyanov, “Combined 2d and 3d tracking of surgical instruments for minimally invasive and robotic-assisted surgery,” *International Journal of Computer Assisted Radiology and Surgery*, pp. 1–11, 2016.

[177] R. Arandjelović and A. Zisserman, “Three things everyone should know to improve object retrieval,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[178] J. Y. Bouguet, “Pyramidal implementation of the lucas kanade feature tracker,” *Intel Corporation, Microprocessor Research Labs*, 2000.

[179] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, pp. 91–110, Nov. 2004.

[180] D. Kondermann, S. Abraham, G. Brostow, W. Förstner, S. Gehrig, A. Imiya, B. Jähne, F. Klose, M. Magnor, H. Mayer, *et al.*, “On performance analysis of optical flow algorithms,” in *Outdoor and Large-Scale Real-World Scene Analysis*, pp. 329–355, Springer, 2012.

[181] B. D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” in *International joint conference on Artificial intelligence*, pp. 674–679, Morgan Kaufmann Publishers Inc., 1981.

[182] J. Shi and C. Tomasi, “Good features to track,” in *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR ’94., 1994 IEEE Computer Society Conference on*, pp. 593–600, Jun 1994.

[183] P.-L. Chang, D. Stoyanov, A. J. Davison, and P. E. Edwards, “Real-time dense stereo reconstruction using convex optimisation with a cost-volume for image-guided robotic surgery,” in *Medical Image Computing and Computer-Assisted Intervention*, pp. 42–49, Springer Berlin Heidelberg, 2013.

[184] R. A. Newcombe, D. Fox, and S. M. Seitz, “Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time,” in *Computer Vision and Pattern Recognition*, pp. 343–352, 2015.

[185] R. A. Newcombe, A. J. Davison, S. Izadi, P. Kohli, O. Hilliges, J. Shotton, D. Molyneaux, S. Hodges, D. Kim, and A. Fitzgibbon, “Kinectfusion: Real-time dense surface mapping and tracking,” in *International Symposium on Mixed and Augmented Reality*, pp. 127–136, 2011.

[186] D. Tang, T.-H. Yu, and T.-K. Kim, “Real-time articulated hand pose estimation using semi-supervised transductive regression forests,” in *International Conference on Computer Vision*, December 2013.

[187] M. W. Spong and M. Vidyasagar, *Robot dynamics and control*. John Wiley & Sons, 2008.

[188] J. J. Craig, *Introduction to robotics: mechanics and control*, vol. 3. Pearson Prentice Hall Upper Saddle River, 2005.

[189] J. Denavit and R. S. Hartenberg, “A kinematic notation for lower-pair mechanisms based on matrices,” *Trans. ASME, J. Appl. Mech.*, vol. 22, no. 2, pp. 215 – 221, 1965.

[190] A. Bartoli and P. Sturm, “Structure-from-motion using lines: Representation, triangulation, and bundle adjustment,” *Computer vision and image understanding*, vol. 100, no. 3, pp. 416–441, 2005.

[191] K. Pachtrachai, M. Allan, and D. Stoyanov, “Hand-eye calibration for robotic assisted minimally invasive surgery without a calibration grid,” in *Intelligent Robots and Systems*, 2016.

[192] P. Pratt, A. Hughes-Hallett, L. Zhang, N. Patel, E. Mayer, A. Darzi, and G.-Z. Yang, “Autonomous ultrasound-guided tissue dissection,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 249–257, Springer, 2015.

[193] Z. Kalal, K. Mikolajczyk, and J. Matas, “Tracking-learning-detection,” *Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, pp. 1409–1422, July 2012.